

By Smriti Sharan ([sfdcamlified](#))

# In Depth Interview Questions on Streaming API, Platform Event and Change Data Capture

*Hello, my Name is Smriti Sharan. I am [avid blogger](#) and [youtuber](#). Follow my Blog and youtube to learn various aspect of Salesforce*

<https://t.me/sfdcamlified> feel free to join telegram group

## Interview Questions on Platform Events

### 1. What is the difference between polling technology and push technology?

Polling technology involves clients periodically requesting data from the server to maintain data freshness. This can lead to unnecessary API calls and potential server slowdowns since the server must respond to each request, even if no data changes occur.

Push technology, on the other hand, involves the server only sending data to the client when there is a change. The client establishes a connection with the server, and the server responds only when there's a data change, ensuring more efficient data handling.



By Smriti Sharan ([sfdcamlified](#))

Example: In an application, if you use polling technology, the client's application might check every minute for changes in order status. This could result in many unnecessary API calls. With push technology, the **server notifies the client immediately when the order status changes, reducing unnecessary traffic and ensuring timely updates.**

## 2. What is Streaming API in Salesforce?

Streaming API in Salesforce enables the transmission of **real-time data from the Salesforce platform**. It is an outbound process where events (changes in data) are sent to external applications. It uses **push technology** to transfer information from the server to the client and **supports multiple subscription methods, such as CometD and the Subscription API.**

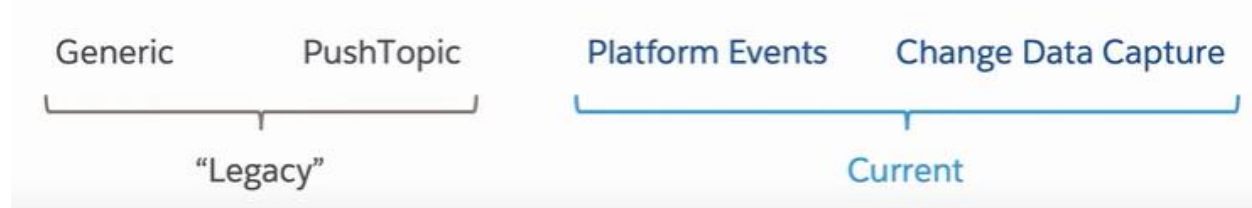
## 3. What are the different types of streaming events available in Salesforce?

**PushTopic Events:** Used to receive notifications for **changes in Salesforce records** based on a defined SOQL query. If a company wants to **be notified only when opportunities in Salesforce are closed**, they can use a **PushTopic event** with a SOQL query that filters for closed opportunities.

**Change Data Capture (CDC) Events:** **Captures all changes to Salesforce records**, including create, update, delete, and undelete operations, without needing a specific SOQL query. If we need to **track all changes to account records**, we should use **Change Data Capture (CDC) events**.

**Generic Events:** Allows sending **custom notifications** from Salesforce to external systems.

**Platform Events:** **Custom events defined by users** to communicate changes and custom notifications. If a company needs to **notify an external order management system whenever a new order is placed** in Salesforce, they can use a Platform Event.



By Smriti Sharan ([sfdcamlified](#))

	PushTopic	Generic	Platform Events	Change Data Capture
Data structure	Defined by target object via SOQL	Unstructured (string)	Custom typed fields	Defined by target object
Publication	Automated	REST API	Declarative or Code	Automated
Subscription	Code		Declarative or Code	

#### 4. What is a Push Topic in Salesforce?

A Push Topic enables streaming of Salesforce record changes to clients based on criteria defined in a SOQL query. It sends notifications for create, update, delete, and undelete operations when records match the criteria specified in the query.

Example: If a business wants to receive notifications every time a case is closed in Salesforce, they can create a Push Topic with a SOQL query that specifies cases with a status of "Closed". The external system subscribed to this Push Topic will receive real-time notifications whenever a case is closed.

#### 5. How do you create a Push Topic in Salesforce?

A Push Topic is created by inserting a record into the PushTopic object in Salesforce. You define a SOQL query and specify the criteria for notifications.

```
PushTopic pushTopic = new PushTopic();
```

```
pushTopic.Name = 'ClosedCases';
```

```
pushTopic.Query = 'SELECT Id, Subject, Status FROM Case WHERE Status = \'Closed\'';
```

```
pushTopic.ApiVersion = 52.0;
```

## By Smriti Sharan ([sfdcamlified](#))

```
pushTopic.NotifyForFields = 'Referenced';  
pushTopic.NotifyForOperationCreate = true;  
pushTopic.NotifyForOperationUpdate = true;  
pushTopic.NotifyForOperationDelete = true;  
pushTopic.NotifyForOperationUndelete = true;  
insert pushTopic;
```

This Push Topic will notify clients about create, update, delete, and undelete operations for cases with a status of "Closed".

### 6. What is the significance of the replay ID in Salesforce Streaming API?

The **replay ID is used to track and retrieve past events in the Streaming API**. It ensures that clients can replay missed events by specifying the last received replay ID when re-establishing a connection.

### 7. What are Generic Events and when should they be used?

Generic Events are used to send notifications that are not tied to Salesforce data changes. They are useful for sending general messages or alerts.

Example: In a system maintenance scenario, a Generic Event can be used to **notify all users that the system will be down for maintenance from 10 PM to 12 AM.**

### 8. How do you publish and subscribe to Generic Events?

Generic Events are published using the REST API and can be **subscribed to using the CometD protocol.**

POST

/services/data/v48.0/subjects/StreamingChannel/streamingChannelId/StreamingChannelPush

**By Smriti Sharan ([sfdcamlified](#))**

```
{  
  "payload": "{\"message\":\"System maintenance from 10 PM to 12 AM\"}"  
}
```

**To subscribe to a Generic Event:**

```
const cometd = new org.cometd.CometD();  
cometd.configure({ url: '/cometd/48.0' });  
cometd.handshake(handshakeReply => {  
  if (handshakeReply.successful) {  
    cometd.subscribe('/event/GenericEvent__e', message => {  
      console.log('Generic Event received: ', message);  
    });  
  }  
});
```

**9. Your organization needs to send general notifications to users whenever a specific business event occurs, such as a system downtime alert or a reminder for a scheduled maintenance. Would you use Platform Events or Generic Events for this purpose?**

For sending general notifications, such as system downtime alerts or maintenance reminders, we would use Generic Events as they are designed for broadcasting simple notification **without requiring a predefined schema.**

Platform Events would be more appropriate **if the notifications were related to specific business data changes.**

**10. You need to notify an internal team within Salesforce whenever a high-priority case is created or updated. The notifications should only be sent to users who have access to the case. Would you use PushTopic Events or Platform Events?**

We would use PushTopic Events because:

## By Smriti Sharan ([sfdcamlified](#))

- PushTopic Events use SOQL queries to define the criteria for event generation.
- PushTopic Events respect Salesforce's sharing model and user permissions.
- PushTopic Events provide real-time updates, which are suitable for notifying internal teams about important case changes as they happen.

## Interview Questions on Platform Events

1. What are platform events in Salesforce?

Platform events are a **product of the streaming API** in Salesforce used to **send and receive custom event notifications**. They are based on an event-driven architecture, where a **publisher sends an event message that subscribers can receive and process**. Platform Events are an example of a **"Fire and Forget" integration**, like outbound messaging, and have a similar look and feel to an sObject or a custom object.



Think of platform events like a **broadcast radio signal** where the station (event producer) sends out a signal (event message) that any radio tuned to the correct frequency (subscribers) can receive.

By Smriti Sharan ([sfdcamlified](#))



Example: Imagine an e-commerce platform like Amazon where a customer places an order. The order placement triggers a platform event that notifies the billing system to generate an invoice, and the shipping system to prepare for delivery.



When creating a Platform event, note the API name suffix ‘\_\_e’, which makes it different from custom objects ( ‘\_\_c’).

## 2. How do platform events differ from Change Data Capture (CDC)?

Platform Events: Custom events defined and published manually as needed. They are useful for scenarios where DML operations are not involved.



## By Smriti Sharan ([sfdcamlified](#))

Example: Notifying the sales team when a user inquires about a product without making a purchase.

Change Data Capture (CDC): Automatically captures changes (create, update, delete) in Salesforce records.

Example: Notifying an external system when an order is created, updated, or deleted in Salesforce.

Feature / Aspect	Platform Events	Change Data Capture (CDC)
Purpose	Custom event messages for custom processes	Captures changes to Salesforce records
Data Type	Custom event messages (custom objects)	Record changes (create, update, delete, undelete)
Schema Definition	Defined by user with custom fields	Based on existing Salesforce objects and fields
Use Cases	- Integrations with external systems	- Data synchronization between systems
	- Notifications within Salesforce	- Auditing changes
	- Asynchronous processing	- Data replication
Subscribers	- Apex Triggers	- Apex Triggers
	- External systems via CometD	- External systems via CometD
	- Flow and Process Builder	- Flow and Process Builder

### 3. What is the Streaming API in Salesforce?

The Streaming API is a way of sending instant notifications from one system to another using a publisher-subscriber model. The publisher sends notifications, and subscribers receive and process them.

Example: A weather alert system where a central server sends weather updates (events) to multiple subscribed devices (subscribers) in real-time.



By Smriti Sharan ([sfdcamplified](#))



#### 4. How is the Streaming API different from the REST API?

REST API: Based on a **client-server architecture**, it involves sending requests and receiving responses. It is **stateless and closes the connection after the response**.

Example: A mobile app sending a request to a server to fetch user profile information and waiting for a response.



Streaming API: **Maintains a persistent connection**, allowing for long-running requests and **stateful interactions**. Ideal for **one-way notifications without expecting a response**.

Example: A stock trading app receiving real-time updates on stock prices through a **continuous stream of data**.

By Smriti Sharan ([sfdcamlified](#))



## 5. What are the components of an event-driven architecture?

Event Producer: The source generating the event.

Event Consumer: The system receiving and processing the event.

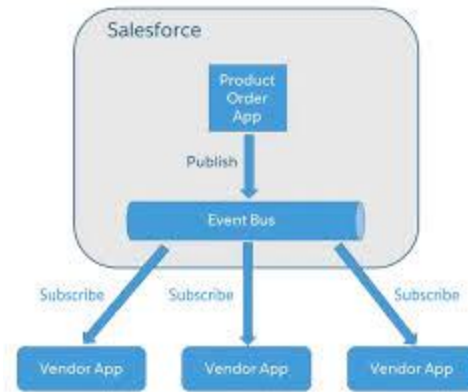
Event Channel: The stream where events are sent and read.

Event Bus: A storage and delivery service for event messages based on a publish-subscribe model.

## 6. Describe a scenario where platform events are used to notify external systems.

When an opportunity is closed in Salesforce, a platform event can notify external vendor apps to process the shipment. Each vendor app subscribes to the event and creates the shipment accordingly.

By Smriti Sharan ([sfdcamlified](#))



## 7. Describe a scenario where platform events are used within Salesforce.

When a lead is reassigned in Salesforce, a platform event can **notify an internal app to reassign the lead and create a Chatter post based on the event information.**

## 8. How do you create a platform event in Salesforce?

Navigate to the Quick Find box, type "Platform Events," and create a new platform event by defining its name and custom fields.

## By Smriti Sharan ([sfdcamplified](#))

The screenshot shows the Salesforce Setup interface. In the left sidebar, the 'Platform Events' option is highlighted under the 'Integrations' section. The main content area displays the 'Platform Events' configuration page. At the top, there is a 'New Platform Event' button. Below it is a table listing existing platform events.

Action	Label	Installed Package	Deployed	Description
<a href="#">Edit</a>   <a href="#">Del</a>	Case Post Event		✓	
<a href="#">Edit</a>	<a href="#">Clause Bundle Selected Event</a>	Conga Contracts	✓	
<a href="#">Edit</a>	<a href="#">Data Update Event</a>	Conga Contracts	✓	
<a href="#">Edit</a>	<a href="#">FDN Internal Event</a>	FinancialForce Foundations	✓	
<a href="#">Edit</a>   <a href="#">Del</a>	New High Priority Case Event		✓	
<a href="#">Edit</a>   <a href="#">Del</a>	New Outage Message		✓	
<a href="#">Edit</a>	<a href="#">Orchestrate Event</a>	Conga Orchestrate	✓	
<a href="#">Edit</a>   <a href="#">Del</a>	Outage Message Date Update		✓	
<a href="#">Edit</a>	<a href="#">Platform Job Event</a>	Conga Contracts	✓	

You can create a platform event named "Order\_Detail\_\_e" with fields like "Order\_Number\_\_c" and "Generate\_Invoice\_\_c" to capture order details and invoice generation status.

### 9. What are the different publish behaviors for platform events?

**Publish After Commit:** The event is published only after the transaction is successfully committed.

An event is published only when an order is successfully created in Salesforce.

**Publish Immediately:** The event is published immediately, regardless of whether the transaction is committed.

An event is published immediately when an order attempt fails, notifying a monitoring system of the failure.

## By Smriti Sharan ([sfdcamlified](#))

### 10. What are replay ID and event UUID in platform events?

Replay ID: A value populated when the event is delivered to the subscriber, used to track the position of the event in the event stream. Example, An order event is delivered to an inventory system, and the replay ID helps track the order's position in the event stream.

Event UUID: A unique identifier for the platform event message, ensuring each event message can be uniquely identified. An order event is uniquely identified using the event UUID to ensure no duplication in processing.

### 11. What limitations exist for platform events?

- Cannot query platform events using SOSL or SOQL.
- Cannot use platform events in reports, list views, or search views.
- Platform event fields are read-only.
- Cannot create tabs for platform events.
- Platform events are unordered and non-replayable with a retention period of 72 hours.

### 12. How can you monitor platform events using the Streaming Monitor app?

1. Install the Streaming Monitor app from the AppExchange.
2. In Salesforce, navigate to the Streaming Monitor app.
3. Subscribe to the platform event channel.
4. View incoming events in real-time as they are published.

### 13. How do platform events handle failures and retries?

## By Smriti Sharan ([sfdcamlified](#))

Platform events are non-replayable, meaning you cannot replay events once they are no longer in the streaming channel. However, subscribers can use the replay ID to track the last received event and handle retries manually if necessary.

Example: If a shipping system fails to process an order event, it can use the replay ID to track the last successful event and request any missed events from the event bus.

### 14. What are the different methods to publish platform events in Salesforce?

Platform events can be published using:

1. Flows - Example: Using a flow to publish a platform event when a new order is created, which triggers notifications to the inventory system.
2. Apex
3. API (SOAP, REST, Bulk)

### 15. How do you publish platform events using Apex?

1. Define the platform event and its fields.
2. Use `EventBus.publish to queue` the event.
3. Validate if the event is queued successfully using `Database.SaveResult`

```
List<Order_Notification__e> eventList = new List<Order_Notification__e>();  
Order_Notification__e orderEvent = new  
Order_Notification__e(Order_Number__c='12345',  
Generate_Invoice__c=true);  
eventList.add(orderEvent);  
Database.SaveResult sr = EventBus.publish(eventList);  
if (sr.isSuccess()) {
```

## By Smriti Sharan ([sfdcamlified](#))

```
System.debug('Event published successfully.');
```

```
} else {
```

```
System.debug('Failed to publish event: ' + sr.getErrors());
```

```
}
```

### 16. How do you publish platform events using the REST API?

1. Send a POST request to /services/data/vXX.X/subjects/EventName\_\_e
2. Include the event data in the request body.

Example: Using Postman to publish an Order\_Notification\_\_e event.

```
{
```

```
  "Order_Number__c": "12345",
```

```
  "Generate_Invoice__c": true
```

```
}
```

### 17: How can you verify the platform events are published successfully?

A8: Use the **Streaming Monitor app from AppExchange** to subscribe to the platform event channel and view incoming events in real-time.

Example: After publishing an event, you can see it listed in the Streaming Monitor with the details like Order\_Number\_\_c and Generate\_Invoice\_\_c.



**By Smriti Sharan ([sfdcamplified](#))**

**18. What are the benefits of using platform events over other integration methods?**

Real-Time Processing: Immediate notifications and actions.

Decoupling: Systems do not need direct dependencies on each other.

Scalability: Multiple subscribers can listen to the same event.

**19. How can you ensure that a Platform Event is successfully published?**

To ensure a Platform Event is successfully published, you can implement `EventBus.PublishSuccessCallback` interface to handle successful publications

`EventBus.PublishFailureCallback` interface to handle failed publications.

`Test.getEventBus().fail()` method to simulate failures for testing purposes.

**20. Can you explain the role of the `Test.getEventBus().fail()` method in testing Platform Events?**

The `Test.getEventBus().fail()` method is used to simulate the failure of a Platform Event publication during test execution. This allows developers to **test the failure handling logic and ensure the system behaves correctly** when an event fails to publish.

**21. How does the `EventBus.PublishCallback` interface help in real-time error handling?**

The `EventBus.PublishCallback` interface helps in real-time error handling by providing methods to handle both successful (`onSuccess`) and failed (`onFailure`) event publications.

This ensures that any issues during the publish process can be caught and managed appropriately.

By Smriti Sharan ([sfdcamlified](#))

Example: In a healthcare application, when an appointment is booked, an event is published. Using the `EventBus.PublishCallback` interface, if the event fails to publish, the system can immediately notify the user to **reattempt the booking** or contact support, ensuring no appointments are missed.

## 22. What are the different ways to subscribe to Platform Events in Salesforce?

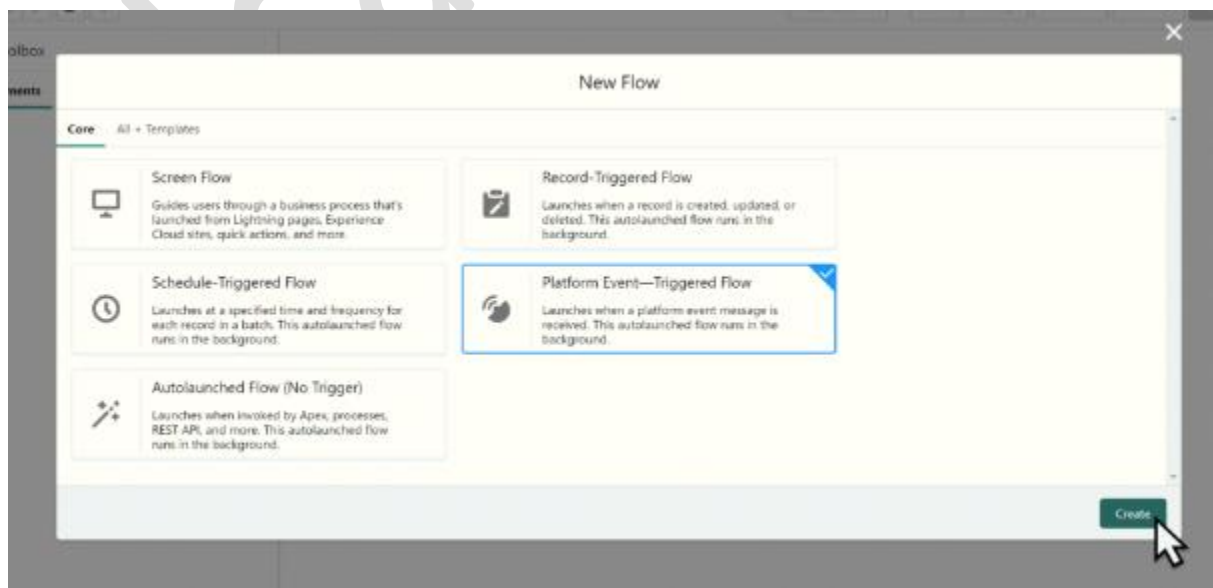
In Salesforce, you can subscribe to Platform Events using:

- Flows
- Apex Triggers
- Lightning Web Components (LWC)

## 23. How do you subscribe to Platform Events using Flows in Salesforce?

To subscribe to Platform Events using Flows:

- Create a new Flow and select the type "Platform Event-Triggered Flow".



- Choose the Platform Event you want to subscribe to.

## By Smriti Sharan ([sfdcamlified](#))

- Define the actions to take place when the event is received, such as creating records or sending notifications.

### 24. What are the key points to consider when subscribing to Platform Events using Apex Triggers?

- The trigger must be on the **after insert event**.
- The default user context for the trigger is the **"Automated Process"** user. This user has specific permissions and does not have a real user account that you can log into.

```
trigger MyPlatformEventTrigger on MyPlatformEvent__e (after insert) {  
    for (MyPlatformEvent__e event : Trigger.New) {  
        // Process the event  
        System.debug('Received Platform Event: ' + event);  
    }  
}
```

### 25. How can Lightning Web Components (LWC) be used to subscribe to Platform Events?

To subscribe to Platform Events using LWC:

- Import the **lightning/empApi module**. This module provides the Event Messaging Platform (EMP) API, which allows you to subscribe to and receive platform events in real-time.
- Use the subscribe method from the module to subscribe to the Platform Event channel.
- Handle the event within the component to perform the desired actions.

```
import { LightningElement, track } from 'lwc';
```

## By Smriti Sharan ([sfdcamlified](#))

```
import { subscribe, onError, setDebugFlag, isEmpEnabled } from  
'lightning/empApi';
```

### 26. What are some common methods provided by the lightning/empApi module in LWC?

The lightning/empApi module provides methods such as:

subscribe(): Subscribes to a Platform Event channel.

unsubscribe(): Unsubscribes from a Platform Event channel.

onError(): Handles errors during subscription.

setDebugFlag(): Enables or disables debug logging.

isEmpEnabled(): Checks if EMP API is enabled.

### 27. How do you handle user context when subscribing to Platform Events using Flows or Triggers?

By default, Platform Event-triggered Flows and Triggers run under the "Automated Process" user. To handle user context:

In Flows

Step 1: Create a new Flow and set it to be triggered by a Platform Event.

Step 2: Add a Get Records element to fetch the user record you want to set as the owner.

Step 3: Use the Create Records or Update Records element to set the fetched user as the owner of the created or modified records.

In Apex Triggers

- In your trigger, query the desired user record.
- Set this user as the owner or "Created By" user for any records created within the trigger.

```
newRecord.OwnerId = targetUser.Id;
```

## 28. How can you test the functionality of subscribing to Platform Events using Apex Triggers?

- Create test Platform Events and publish them in a test context.
- Use the `Test.getEventBus().deliver()` method to simulate event delivery. This method is used to simulate the delivery of platform events.
- Assert the expected outcomes, such as the creation of records or tasks.

## 29. What is the EMP API module in LWC and its limitations?

The lightning/empApi module in LWC provides methods to subscribe to and unsubscribe from streaming events such as Platform Events. It supports methods like subscribe, unsubscribe, onError, setDebugFlag, and isEmpEnabled.

Limitations:

The lightning/empApi module is only supported for desktop applications and is not available for the Salesforce mobile app.

By Smriti Sharan ([sfdcamlified](#))

**30. How do you ensure a LWC subscribes to a Platform Event only when it is loaded and unsubscribes when it is unloaded?**

To manage subscriptions correctly:

1. Use the `connectedCallback` lifecycle hook to subscribe to the Platform Event when the component is loaded.
2. Use the `disconnectedCallback` lifecycle hook to unsubscribe from the Platform Event when the component is unloaded.

**31. How can you handle errors while subscribing to Platform Events in LWC?**

Handle errors by using the `onError` method from the `lightning/empApi` module to set an error handler function.

**32. How do you notify users about the successful update of a record when a Platform Event is received in LWC?**

Use the `ShowToastEvent` from [lightning/platformShowToastEvent](#) to notify users about the successful update.

```
import { ShowToastEvent } from 'lightning/platformShowToastEvent';
```

By Smriti Sharan ([sfdcamplified](#))

### 33. What is the purpose of implementing the Database.RaisesPlatformEvents interface in a Batch Apex class?

The Database.RaisesPlatformEvents interface is used to automatically fire a platform event whenever an error or exception occurs in a Batch Apex job. This allows for efficient error handling by subscribing to these platform events and taking appropriate actions.

```
public with sharing class SampleBatchJob implements
    Database.Batchable<SObject>,
    Database.RaisesPlatformEvents{

    // class implementation
}
```

### 34. What is the difference between platform events and subjects?

Although Platform Events have the look and feel of an sObject, the biggest difference is that Platform Events are immutable (they can't be updated once published).

### 35. Explain Event-Driven Architecture in the context of Salesforce?

Event-driven architecture involves event producers broadcasting messages to an event bus, which holds and stores the messages until they are consumed by event consumers. In Salesforce, platform events can hold messages for 24 hours, while change data capture events can hold messages for up to 72 hours.



## By Smriti Sharan ([sfdcamlified](#))

Example: When an order is placed in Salesforce (event producer), an event is **broadcasted to the event bus**. The inventory management system (event consumer) receives the event and updates the stock levels accordingly.

### 36. How to create a Platform Event and subscribe to it using Lightning Web Components (LWC)?

#### To create a Platform Event:

- Navigate to Setup > Platform Events > New Platform Event.
- Provide a name and select the appropriate publishing behavior (e.g., "Publish after commit").
- Create custom fields as needed (e.g., a text field named "Message").

#### To subscribe using LWC:

Use the **provided EMP API to subscribe** to the event:

```
const cometd = new org.cometd.CometD();

cometd.configure({ url: '/cometd/48.0' });

cometd.handshake(handshakeReply => {
  if (handshakeReply.successful) {
    cometd.subscribe('/event/YourPlatformEvent__e', message => {
      console.log('Platform Event received: ', message);
    });
  }
});
```

By Smriti Sharan ([sfdcamlified](#))

### 37. Can you write triggers for Platform Events?

Yes, you can write triggers for both Platform Events and Change Data Capture events.

```
trigger OrderStatusChangeTrigger on OrderStatusChange__e (after insert)
{
    for (OrderStatusChange__e event : Trigger.New) {
        // Process the event

        System.debug('Order ID: ' + event.OrderId__c);

        System.debug('Status: ' + event.Status__c);
    }
}
```

### 38. Your company needs to synchronize account data between Salesforce and a third-party CRM system in **real-time**. Should you use Platform Events or PushTopic Events for this integration?

For real-time account data synchronization, we would use Platform Events instead of PushTopic Events because:

1. **Customizable Event Payloads:** Platform Events allow for custom event payloads, enabling us to **define specific fields and data that need to be synchronized** with the third-party CRM system.
2. **Asynchronous Processing:** Platform Events are designed for high-volume, asynchronous processing.
3. **Publish and Subscribe Model:** Platform Events support the publish-subscribe model. Multiple subscribers can listen to the events without affecting each other.

By Smriti Sharan ([sfdcamlified](#))

4. **No SOQL Query Limitations:** Unlike PushTopic Events, which rely on SOQL queries and have limitations on the query complexity and number of fields, Platform Events do not have such restrictions.

PushTopic Events are better suited for scenarios where you need to track changes based on specific SOQL queries and when the data volume is relatively low.

## Interview Questions on Change Data Capture

### 1. What is Change Data Capture (CDC) in Salesforce?

Change Data Capture (CDC) is a streaming product in Salesforce that enables the real-time synchronization of Salesforce data changes (create, update, delete, and undelete) with external systems. It uses the Streaming API to notify external systems of data changes.

#### When to use it:

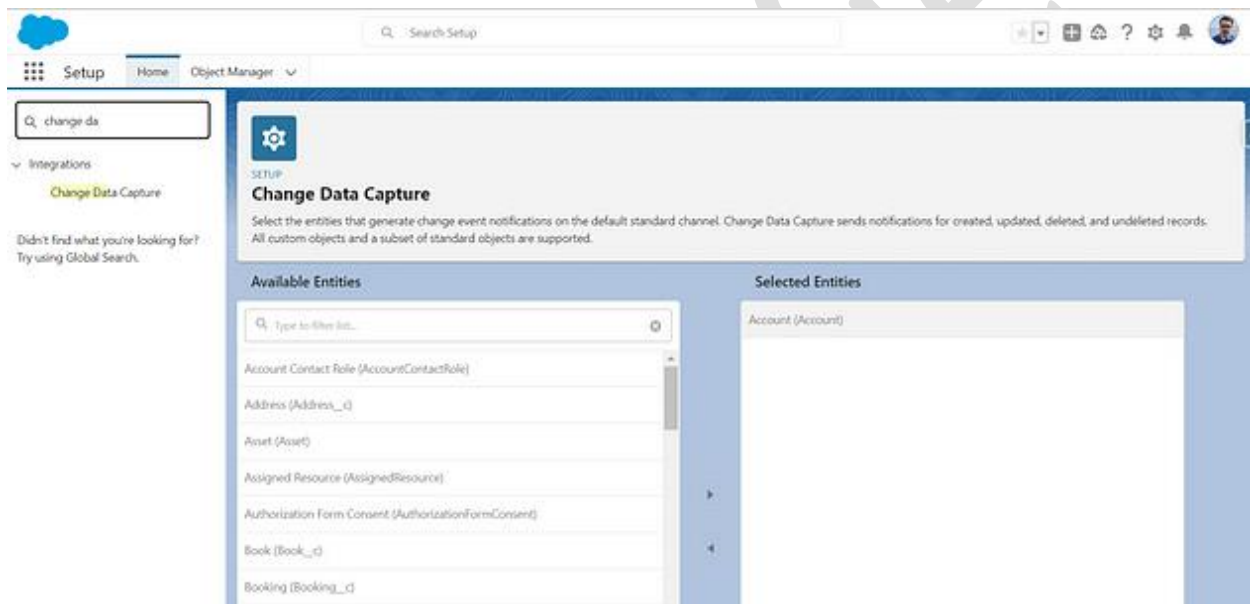
- Receive notifications of Salesforce record changes, including create, update, delete, and undelete operations.
- Capture all field changes for all records.
- Get broad access to all data regardless of sharing rules.
- Get information about the change in the event header, such as the origin of the change, so you can ignore changes that your client generates.
- Perform data updates using transaction boundaries when more than one operation is part of the same transaction.
- Use a versioned event schema.

## By Smriti Sharan ([sfdcamlified](#))

- Subscribe to mass changes in a scalable way.
- Get access to retained events for up to 3 days.

### 2.How to enable CDC?

Go to Setup, In Quick Find box, under integration you will see 'Change Data Capture' then you can select whatever object (entities) you want to perform then click save button.



### 3. How to Subscribe CDC using ASYNCHRONOUS APEX TRIGGER?

To subscribe CDC you can use Asynchronous Apex Trigger which is similar like how we subscribe Platform Event using trigger.

**Consideration :-**

## By Smriti Sharan ([sfdcamlified](#))

- For a standard object, the change event name is StandardObjectNameChangeEvent for example AccountChangeEvent
- For a custom object, the change event name is CustomObjectName\_\_ChangeEvent for example Employee\_\_ChangeEvent
- Support only after insert.

### Syntax

```
trigger CDCDemo on AccountChangeEvent (after insert) {  
  
}
```

### 4.How to Subscribe CDC using LWC

To use the **empApi** methods in your Lightning web component, import the methods from the **lightning/empApi** module as given below.

```
import { subscribe, unsubscribe, onError, setDebugFlag, isEmpEnabled  
from 'lightning/empApi'; }
```

### 5. What are the main components of a CDC payload in Salesforce?

entityName: The name of the entity (object) that changed.

recordIds: Array of record IDs affected by the change.

**By Smriti Sharan ([sfdcamlified](#))**

changeType: The type of change (create, update, delete, undelete).

transactionKey and sequenceNumber: Identifiers for the transaction and sequence of changes.

fields: The specific fields that were changed.

## **6. What are the different types of changes that CDC can capture, and how is the data sent?**

CDC can capture the following types of changes:

- Create
- Update
- Delete
- Undelete

The data sent by CDC is only the changed fields are included in the notification, not the entire record.

Example: When an Account record's phone number and address are updated, the CDC payload includes only the changed fields (phone number and address) rather than the entire Account record. This reduces bandwidth and processing overhead.

## **7. What is a custom channel in Change Data Capture (CDC) in Salesforce, and why would you use it?**

A custom channel in CDC allows developers to create specific channels for different subscribers or use cases.

## By Smriti Sharan ([sfdcamlified](#))

Example:

An organization integrates Salesforce with both an ERP system and a billing system. The ERP system needs data from Account, Contact, and Opportunity objects, while the billing system requires data from Product, Quote, and Order objects. Using custom channels, the organization creates two separate channels to ensure each system only receives relevant data.

### 8. How do you create a custom channel in CDC, and what are the main components required?

To create a custom channel in CDC, you need to use the Metadata API or Tooling API. The main components required are:

PlatformEventChannel: Defines the custom channel with attributes like FullName, ChannelType, and Label.

PlatformEventChannelMember: Adds specific objects to the custom channel, with attributes like FullName, EventChannel, and SelectedEntity.

### 9. What is event enrichment in CDC, and how do you implement it?

Event enrichment in CDC allows adding additional fields to the events, beyond the fields that triggered the event. This is useful for providing more context to the receiving system.

### 10. How do you filter events in a custom channel, and what are the benefits of doing so?



**By Smriti Sharan ([sfdcamlified](#))**

Filtering events in a custom channel involves defining filter expressions that specify conditions under which events are sent. This reduces unnecessary data transfer and processing.

Only send Account updates to the ERP system if the Account's industry is "Energy". This ensures the ERP system only processes relevant data.

Add a filter expression to the `PlatformEventChannelMember`:

```
{  
  "FullName": "ERP_Channel_Account_Member",  
  "Metadata": {  
    "FilterExpression": "Industry = 'Energy'",  
    "EventChannel": "ERP_Channel",  
    "SelectedEntity": "AccountChangeEvent"  
  }  
}
```

## **11.How do you create a Change Data Capture trigger in Salesforce, and what is its structure?**

To create a Change Data Capture trigger, you need to define an Apex trigger on the change event object of the entity you want to monitor. The trigger should always run in the after insert context.

## By Smriti Sharan ([sfdcamlified](#))

Example, For a Contact object, you would create a trigger on ContactChangeEvent and handle the logic to respond to changes in Contact records.

```
trigger ContactChangeEventTrigger on ContactChangeEvent (after insert) {  
    for (ContactChangeEvent event : Trigger.New) {  
        // Handle the change event  
    }  
}
```

### 12.What are the limitations of Change Data Capture triggers in Salesforce?

- CDC triggers only run in the **after insert context**.
- They operate asynchronously after the main transaction commits.
- The default owner of records created by CDC triggers is the Automated Process user.
- The maximum batch size for event messages in CDC triggers is 2,000.

### 13. What are the key methods used in writing test classes for CDC triggers?

Test.enableChangeDataCapture(): Enables the generation of change event notifications in the test context.

**By Smriti Sharan ([sfdcamlified](#))**

`Test.getEventBus().deliver()`: Delivers the change event notifications.

`Test.startTest()` and `Test.stopTest()`: Used to encapsulate the test logic and deliver the change events.

#### **14. How can you subscribe to Change Data Capture events using Lightning Web Components (LWC)?**

To subscribe to CDC events using LWC, you can use the EMP API module provided by Salesforce. This involves importing the necessary methods from the EMP API, subscribing to the relevant change event channels, and handling the change event responses.

Steps:

- Import the EMP API methods in your LWC JavaScript file.
- Subscribe to the CDC event channel in the `connectedCallback` method.
- Handle the change event responses to update your component's state or UI.

#### **15. When to use platform event vs CDC?**

Use Platform Events when you need to send custom notifications that are not directly tied to changes in Salesforce records. They are great for integrating Salesforce with external systems and creating event-driven architectures.

## By Smriti Sharan ([sfdcamlified](#))

Example: You have a Salesforce application where sales representatives place orders, and you need to notify an external system to fulfill these orders.

Change Data Capture (CDC): Use CDC when you need to track changes to Salesforce records (such as create, update, delete, or undelete operations) and synchronize these changes with external systems.

Example: You want to keep an external CRM system in sync with Salesforce whenever changes occur in Salesforce Account, Contact records.

### 16. When should you use Change Data Capture (CDC) over Push Topic?

Use CDC when:

- You need to capture all changes to Salesforce records (create, update, delete, undelete) without specific criteria.
- You need real-time synchronization of data with external systems.
- You want to avoid the complexity of writing and maintaining SOQL queries.

Example:

An external CRM system that needs to maintain a real-time copy of Salesforce account records should use CDC. This ensures that any change to account records in Salesforce is immediately reflected in the external CRM system.

### 17. What are some use cases for Push Topic and Change Data Capture (CDC)?

## By Smriti Sharan ([sfdcamlified](#))

### Push Topic Use Cases:

- When specific criteria need to be met for notifications (e.g: notify when cases are closed).
- When notifications are needed for specific field changes based on a SOQL query.

Example: An external customer support application needs to be notified only when a case is closed in Salesforce. A Push Topic with a SOQL query filtering cases with a status of "Closed" can be used.

### Change Data Capture (CDC) Use Cases:

- When all changes to an object need to be captured without specific criteria.
- When real-time synchronization with external systems is required.

Example: A system needs to stay in sync with all changes to account records in Salesforce.

## 18. Can you write triggers for Change Data Capture events?

Yes, you can write triggers for both Platform Events and Change Data Capture events.

```
trigger ContactChangeTrigger on ContactChangeEvent (after insert) {  
    for (ContactChangeEvent event : Trigger.New) {  
        // Process the change event  
        if (event.ChangeType == 'UPDATE') {  
            // Handle updates  
            System.debug('Contact updated: ' + event.ContactId);  
        }  
    }  
}
```

By Smriti Sharan ([sfdcamlified](#))

## 19. What are the limitations and challenges of using Salesforce CDC?

Complex Implementation: Capturing changes alone is not enough, integrating with target systems requires **additional custom code**.

Data Storage: CDC data can quickly **consume significant storage space**, requiring careful monitoring and planning.

API Limits: CDC uses API requests, and exceeding API limits can lead to service interruptions.

Supported Objects: **Not all Salesforce objects and fields support CDC**.

Historical Data: **CDC only captures changes made after enabling it**, so historical data is not available.

**20. Developer is tasked with developing an HR app that synchronizes employee data between Salesforce and an external HR system. The app **must handle record creation, updates, and deletions in real-time**. Given the requirements, would you use Salesforce Change Data Capture (CDC) or Platform Events? Explain why.**

We would use Salesforce Change Data Capture (CDC) because:

1. **Real-Time Synchronization:** CDC provides near real-time notifications of data changes.
2. **Complete Record Changes:** CDC captures all changes to the employee records, including create, update, delete, and undelete operations.
3. **Minimal Custom Coding:** Using CDC reduces the amount of custom code as it **automatically publishes change events for the selected objects**.
4. **Replayability:** CDC events are **stored for up to three days, allowing the external system to retrieve missed events** in case of failures or downtime.

**By Smriti Sharan ([sfdcamlified](#))**

## About Me

*Hello, my Name is Smriti Sharan. I am [avid blogger](#) and [youtuber](#). Follow my Blog and youtube to learn various aspect of Salesforce*

<https://t.me/sfdcamlified> feel free to join telegram group

## References:

[Salesforce Developer Guide: Testing Platform Event Triggers](#)

[Salesforce Apex Developer Guide: Testing Asynchronous Apex](#)