**What is AWS SNS Delivery Retry?**

**AWS SNS Delivery Retry** is a feature that ensures your messages are delivered even if the first attempt fails. If a message cannot be delivered to a subscriber (e.g., due to a network issue or a server being down), SNS will **automatically retry sending the message** multiple times until it succeeds or reaches the maximum retry limit.

**Why is Delivery Retry Important?**

1. **Reliability**: It ensures that your messages are delivered even if there are temporary issues.
2. **No Lost Messages**: Prevents messages from being lost due to failures.
3. **Better User Experience**: Subscribers receive important notifications without delays.

**How Does Delivery Retry Work?**

1. **First Attempt**: SNS tries to deliver the message to the subscriber.
2. **If Delivery Fails**: SNS waits for a short time and tries again.
3. **Retry Attempts**: SNS will keep retrying for a **specific number of times** (retry limit) or until the message is successfully delivered.
4. **Retry Intervals**: The time between retries increases gradually (e.g., 1 second, 3 seconds, 10 seconds, etc.).

**Key Terms to Understand**

1. **Retry Policy**: A set of rules that defines how SNS should retry message delivery.
   * **Maximum Retries**: The total number of times SNS will retry.
   * **Retry Intervals**: The time gap between each retry attempt.
2. **Dead-Letter Queue (DLQ)**: If a message cannot be delivered even after all retries, it can be sent to a **Dead-Letter Queue** for further analysis.

**Example of Delivery Retry**

Imagine you are sending a **weather alert** to a subscriber’s email:

1. SNS tries to send the email, but the email server is temporarily down.
2. SNS waits for a few seconds and tries again.
3. If the server is still down, SNS retries a few more times.
4. If the email server comes back online, the message is delivered successfully.
5. If all retries fail, the message is sent to a **Dead-Letter Queue** for review.

**How to Set Up Delivery Retry in AWS SNS**

1. **Create a Topic**: Set up an SNS topic (e.g., "WeatherAlerts").
2. **Add Subscribers**: Add subscribers (e.g., email, SMS, or HTTP endpoints).
3. **Configure Retry Policy**:
   * Go to the **Delivery retry policy** section in the SNS console.
   * Set the **maximum retries** (e.g., 3 retries).
   * Set the **retry intervals** (e.g., 1 second, 3 seconds, 10 seconds).
4. **Enable Dead-Letter Queue (Optional)**:
   * Create an **SQS queue** to act as a Dead-Letter Queue.
   * Link the DLQ to your SNS topic to capture failed messages.

**Benefits of Delivery Retry**

1. **Improved Reliability**: Ensures messages are delivered even in case of temporary failures.
2. **Flexibility**: You can customize the retry policy based on your needs.
3. **Error Handling**: Failed messages can be analyzed using a Dead-Letter Queue.

**When to Use Delivery Retry**

1. **Critical Notifications**: For important alerts that must reach the subscriber (e.g., security alerts, payment reminders).
2. **Unreliable Networks**: When subscribers are on networks that may experience temporary issues.
3. **High-Volume Messaging**: For applications sending a large number of messages where occasional failures are expected.

**Simple Summary**

* **AWS SNS Delivery Retry** ensures your messages are delivered even if the first attempt fails.
* It **retries sending the message** multiple times with increasing delays between attempts.
* If all retries fail, the message can be sent to a **Dead-Letter Queue** for further analysis.
* This feature makes SNS more **reliable** and ensures no important messages are lost.