Objective - The Objective of the project is to configure webhook in razorpay dashboard and configure webhook in aws.

Goal to achieve - Once the payment is successful, Webhook to send payment details to aws & check the payment details in aws lambda cloudwatch.

What is a webhook ?

A **webhook** is a way for one application to **send real-time data** to another application **when a specific event occurs**. It is a **server-to-server notification mechanism** for sending real time Data.

What is an Event ?

Understand from this scenario, lets say the payment is successfully done and the payment gateway captures the payment. Here the **event is payment capture. Based on this event we will call the Callback API and send the data.**

Configuration in AWS

Prerequisite

1. AWS login

Configuration

1. Go to aws lambda >>Create a function >> Paste the code/ script

Make sure (important step) - Mention the correct Secret code, the same secret code should be mentioned in the razorpay configuration. Save and Deploy.

2. Create an API in AWS API Gateway

2.1 Go to AWS API gateway >> create an API >> REST API or just for checking you can create HTTP API.

2.2 Use method - ANY and path name

2.2 In the Integration section >> Select AWS lambda >> Select your function. Save and Deploy.

Important step - You will receive a URL -
https://7bclvqzs76.execute-api.us-east-1.amazonaws.com/default

Your webhook url will be will pathname -
https://7bclvqzs76.execute-api.us-east-1.amazonaws.com/default/razorpayWebhookHandler

Your API name is - with method - ANY

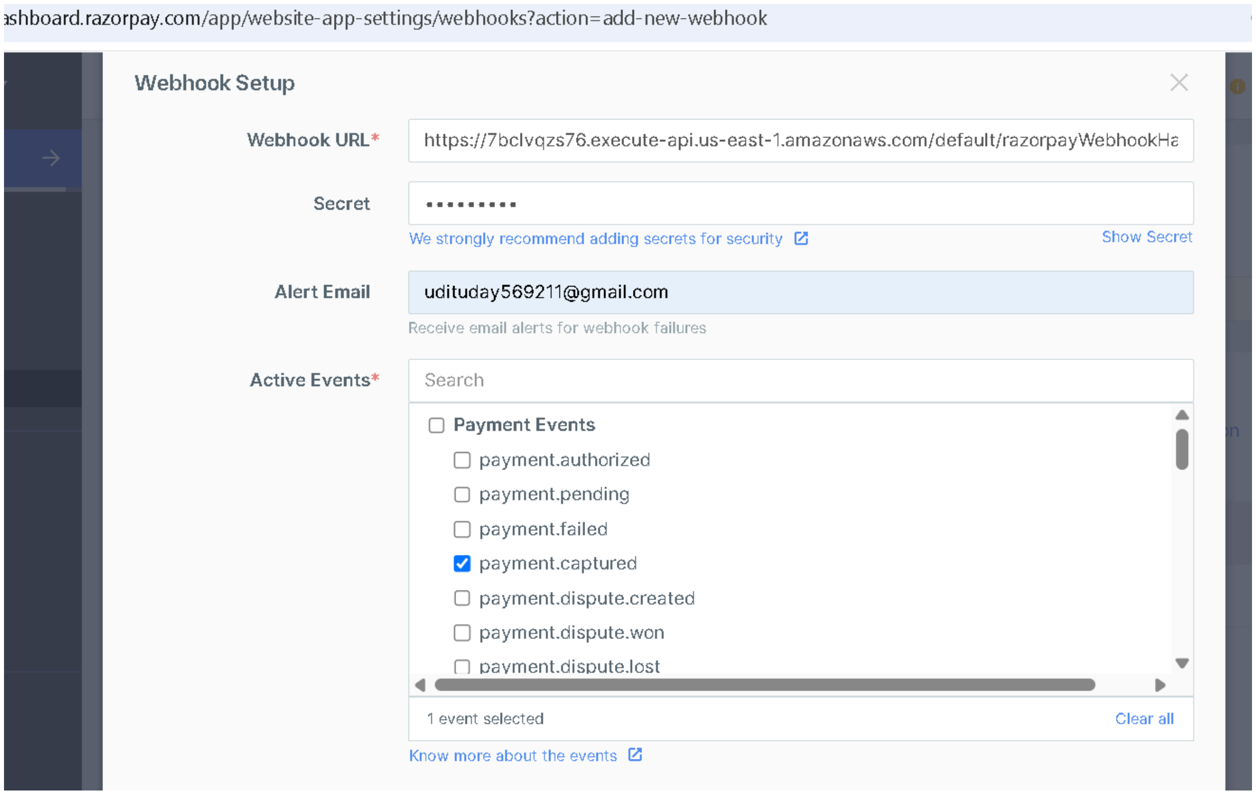
Configuration in Razorpay dashboard?

Prerequisites

1. Razorpay login - Register and do a free login.
2. Go to Account settings >> Click on webhooks.
3. Click on add webhook
4. Type the URL and Secret code
5. Active event >> payment.event
6. Save

Important - Razorpay also does transactions in paisa , so if you are paying **50 rupees ,** actually razorpay is doing it as **5000 paisa.** It is converting and showing you in rupees.

Screenshot for reference -


ashboard.razorpay.com/app/website-app-settings/webhooks?action=add-new-webhook

Webhook Setup

Webhook URL* https://7bclvqzs76.execute-api.us-east-1.amazonaws.com/default/razorpayWebhookHa

Secret ••••••••••

We strongly recommend adding secrets for security — Show Secret

Alert Email udituday569211@gmail.com

Receive email alerts for webhook failures

Active Events*

- [ ] Payment Events
  - [ ] payment.authorized
  - [ ] payment.pending
  - [ ] payment.failed
  - [x] payment.captured
  - [ ] payment.dispute.created
  - [ ] payment.dispute.won
  - [ ] payment.dispute.lost

1 event selected — Clear all

Know more about the events

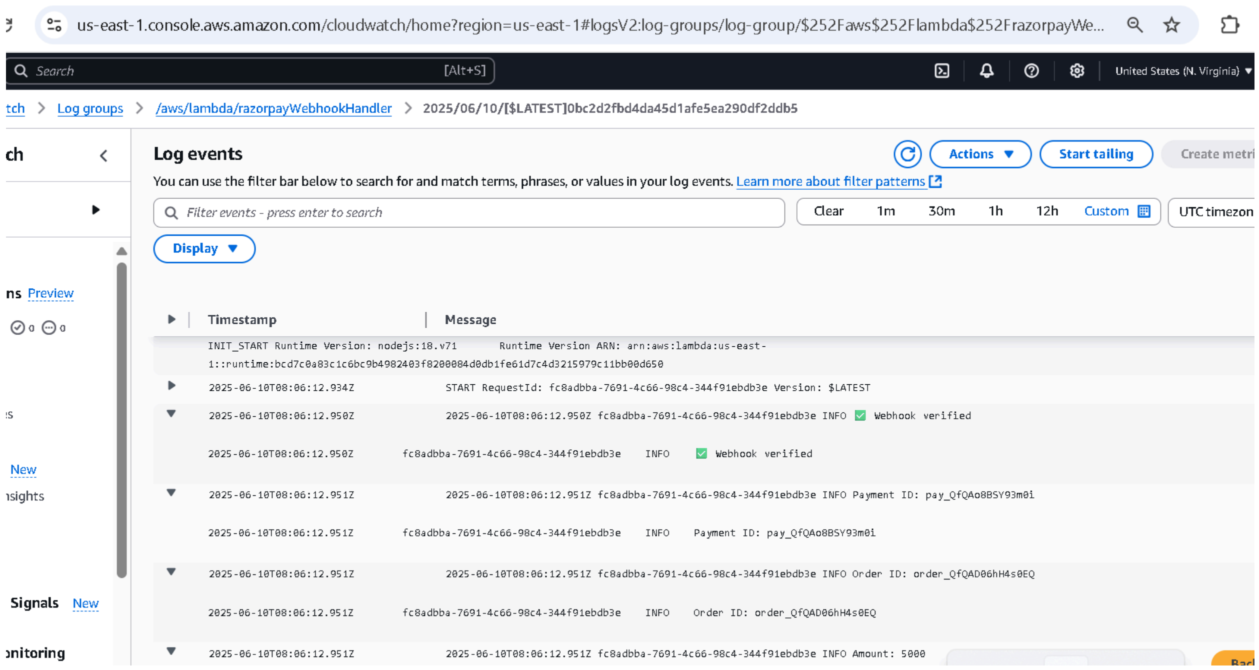
AWS lambada code -

```
const crypto = require("crypto");

exports.handler = async (event) => {
  const webhookSecret = "XXXXXX"; // Your Razorpay webhook secret

  const body = event.body;
  const signature = event.headers['x-razorpay-signature'];

  // Signature verification
  const expectedSignature = crypto
    .createHmac('sha256', webhookSecret)
    .update(body)
    .digest('hex');

  if (signature === expectedSignature) {
    console.log("✅ Webhook verified");

    const payload = JSON.parse(body);
    const paymentData = payload.payload?.payment?.entity;

    if (paymentData) {
      const payment_id = paymentData.id;
      const order_id = paymentData.order_id;
      const amount = paymentData.amount;
      const status = paymentData.status;
      const created_at = new Date(paymentData.created_at * 1000).toISOString();

      console.log("Payment ID:", payment_id);
      console.log("Order ID:", order_id);
      console.log("Amount:", amount);
      console.log("Status:", status);
      console.log("Payment Date:", created_at);
    } else {
      console.log("❌ Payment data not found in webhook payload.");
    }

    return {
      statusCode: 200,
      body: JSON.stringify({ message: "Webhook received and processed." }),
    };
  } else {
    console.log("❌ Invalid webhook signature");
    return {
            statusCode: 400,
            body: JSON.stringify({ message: "Invalid signature" }),
        };
    }
};
```

Result and Observations

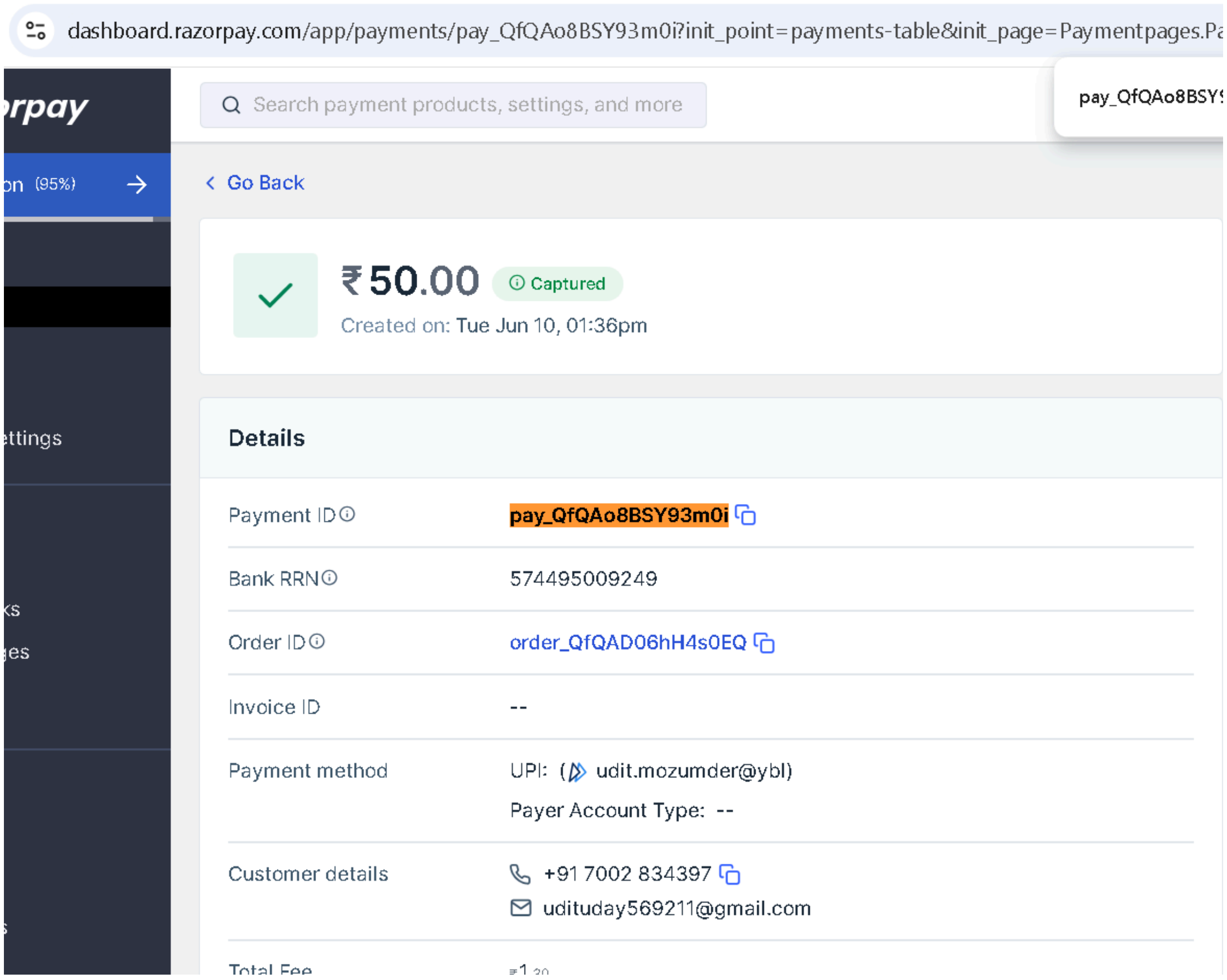
1. Make a successful payment
2. Go to AWS lambda >> Monitor >> Cloudwatch
3. Check the payment id , order id , amount

Screenshot


us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:log-groups/log-group/$252Faws$252Flambda$252FrazorpayWe...
Search [Alt+S]
United States (N. Virginia)
tch > Log groups > /aws/lambda/razorpayWebhookHandler > 2025/06/10/[$LATEST]0bc2d2fbd4da45d1afe5ea290df2ddb5
Log events
Actions
Start tailing
Create metri
Filter events - press enter to search
Clear 1m 30m 1h 12h Custom
UTC timezon
Display
ns Preview
New
ısights
Signals New
onitoring

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                 Message
INIT_START Runtime Version: nodejs:18.v71       Runtime Version ARN: arn:aws:lambda:us-east-
1::runtime:bcd7c0a83c1c6bc9b4982403f8200084d0db1fe61d7c4d3215979c11bb00d650
2025-06-10T08:06:12.934Z          START RequestId: fc8adbba-7691-4c66-98c4-344f91ebdb3e Version: $LATEST
2025-06-10T08:06:12.950Z          2025-06-10T08:06:12.950Z fc8adbba-7691-4c66-98c4-344f91ebdb3e INFO ✅ Webhook verified

2025-06-10T08:06:12.950Z      fc8adbba-7691-4c66-98c4-344f91ebdb3e    INFO    ✅ Webhook verified

2025-06-10T08:06:12.951Z          2025-06-10T08:06:12.951Z fc8adbba-7691-4c66-98c4-344f91ebdb3e INFO Payment ID: pay_QfQAo8BSY93m0i

2025-06-10T08:06:12.951Z      fc8adbba-7691-4c66-98c4-344f91ebdb3e    INFO    Payment ID: pay_QfQAo8BSY93m0i

2025-06-10T08:06:12.951Z          2025-06-10T08:06:12.951Z fc8adbba-7691-4c66-98c4-344f91ebdb3e INFO Order ID: order_QfQAD06hH4s0EQ

2025-06-10T08:06:12.951Z      fc8adbba-7691-4c66-98c4-344f91ebdb3e    INFO    Order ID: order_QfQAD06hH4s0EQ

2025-06-10T08:06:12.951Z          2025-06-10T08:06:12.951Z fc8adbba-7691-4c66-98c4-344f91ebdb3e INFO Amount: 5000
```



dashboard.razorpay.com/app/payments/pay_QfQAo8BSY93m0i?init_point=payments-table&init_page=Paymentpages.Pa

₹50.00 Captured

Created on: Tue Jun 10, 01:36pm

Details

Payment ID	pay_QfQAo8BSY93m0i
Bank RRN	574495009249
Order ID	order_QfQAD06hH4s0EQ
Invoice ID	--
Payment method	UPI: (udit.mozumder@ybl) Payer Account Type: --
Customer details	+91 7002 834397 udituday569211@gmail.com
Total Fee	₹1.20