

## Environment details

- AWS profile used for verification: `rahul-hudle-tech`
- AWS Account ID: `459728651417`
- AWS Region: `ap-south-1`
- Queue name: `rahul_nestjs_notifications-queue`
- Queue URL: `https://sqs.ap-south-1.amazonaws.com/459728651417/rahul_nestjs_notifications-queue`
- Queue ARN: `arn:aws:sqs:ap-south-1:459728651417:rahul_nestjs_notifications-queue`
- Notification job value to use: `PUSH_NOTIFICATION`

## Required IAM permissions

Use least-privilege access. Mandatory permissions are only for dispatch; log permissions are optional and only needed if partner is expected to self-debug.

Permission	Mandatory/Optional	Purpose
<code>sqs:SendMessage</code> on <code>arn:aws:sqs:ap-south-1:459728651417:rahul_nestjs_notifications-queue</code>	Mandatory	Allows partner to publish notification messages to Rahul notification queue.

Recommended default for partners:

- Grant only `sqs:SendMessage` initially.

## Step 1: Configure AWS CLI

```
aws configure
```

Provide:

- Access Key ID (shared by Hudle)
- Secret Access Key (shared by Hudle)
- Region: `ap-south-1`
- Output: `json`

## Step 2: Resolve queue URL

Queue details for this environment:

- Queue name: `rahul_nestjs_notifications-queue`
- Queue URL: `https://sqs.ap-south-1.amazonaws.com/459728651417/rahul_nestjs_notifications-queue`

You can also resolve it via CLI:

```
aws sqs get-queue-url \  
  --region ap-south-1 \  
  --profile rahul-hudle-tech \  
  --queue-name "rahul_nestjs_notifications-queue"
```

Expected `QueueUrl` in response:

```
https://sqs.ap-south-1.amazonaws.com/459728651417/rahul_nestjs_notifications-queue
```

Validate queue attributes:

```
aws sqs get-queue-attributes \  
  --region ap-south-1 \  
  --profile rahul-hudle-tech \  
  --queue-url "https://sqs.ap-south-1.amazonaws.com/459728651417/rahul_nestjs_notifications-queue" \  
  --attribute-names All
```

## Step 3: Prepare message payload

Create `message.json`:

```
{  
  "job": "PUSH_NOTIFICATION",  
  "data": {  
    "users": [493992],  
    "message": {  
      "value": "You have a new update from Hudle Vision."  
    },  
    "notification": {  
      "title": "New Vision Notification",  
      "body": "You have a new update from Hudle Vision.",  
      "image": ""  
    },  
    "type": 0,  
  },  
}
```

```
"meta": {
  "meta_type": 19,
  "actionType": 19
}
}
```

Notes:

- `users` must contain Hudle `user_id` values.
- For phone `9667545342` , resolved user id is `493992` .

## Step 4: Dispatch to SQS

Using `message.json` file:

```
aws sqs send-message \
  --region ap-south-1 \
  --queue-url "https://sqs.ap-south-
1.amazonaws.com/459728651417/rahul_nestjs_notifications-queue" \
  --message-body file://message.json
```

One-liner example:

```
aws sqs send-message --region ap-south-1 --queue-url "https://sqs.ap-
south-1.amazonaws.com/459728651417/rahul_nestjs_notifications-queue" --
message-body '{"job":"PUSH_NOTIFICATION","data":{"users":
[493992],"message":{"value":"You have a new update from Hudle
Vision."},"notification":{"title":"New Vision Notification","body":"You
have a new update from Hudle Vision."},"image":""},"type":0,"meta":
{"meta_type":19,"actionType":19}}}'
```

Expected success response includes:

- `MessageId`
- `MD5ofMessageBody`

## Contract reference

The queue payload follows the same generic push contract used by Vision notifications:

- `job` : `PUSH_NOTIFICATION`
- `data.users` : number array
- `data.message.value` : string
- `data.notification` : { `title`, `body`, `image` }
- `data.type` : number (broadcast is `0` )

- `data.meta` : notification action metadata

## Common failures

- `AccessDenied` on send: IAM policy missing `sqs:SendMessage` on this queue.
- `NonExistentQueue` : wrong queue name/URL or wrong region.
- No delivery to user: verify correct `user_id` and payload format.