Running the Application:

A Step-By-Step Guide

# Pre-requisites:

Terraform, AWS Codepipeline or Jenkins, AWS CodeBuild, S3 bucket and DynamoDB for Terraform backend

Infrastructure Provisioning

### Run Lambda Code Tests:

1. Navigate to “terraform” directory and run the following command to run tests.

* python -m pytest lambda\_python\_code\_tests/

### Steps to Provision through AWS Codepipeline:

1. Configure the CodePipeline to sync with the specified GitHub branch.
2. Establish an AWS CodeBuild project in conjunction with the CodePipeline.
3. Initiate the pipeline, triggering the execution of the buildspec.yaml file.
4. Inside the buildspec.yaml file, you'll find all the commands required for provisioning AWS infrastructure during the build stage.
5. Following the successful completion of the pipeline, AWS services (SNS, Lambda, SQS, DynamoDB, IAM Roles, and Policies) will be generated

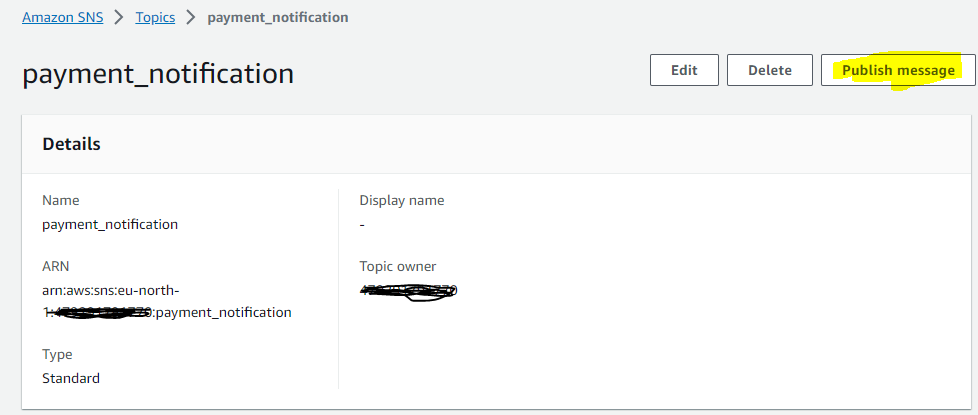
### Steps to Provision through local:

1. Assign a role or include access keys endowed with the necessary permissions (such as creating SNS, Lambda, SQS, DynamoDB, IAM Roles, and Policies) to the local system.
2. Retrieve the code from the "[payment\_application\_notification](https://github.com/umaparvathir/payment_notification_application)" repository.
3. Navigate to the designated Terraform directory and execute the following commands:

* cd terraform
* terraform init
* terraform plan
* terraform apply -var-file=dev\_env\_tf\_scripts.tfvars

Test Application

1. To facilitate testing, SNS will be utilized to push payment notifications. Please proceed to the 'payment\_notification' SNS topic and publish a payment notification.

****

**Example payment notification:**

{

"paymentId": "1234",

"paymentMethod": "paypal",

"timeStamp": "13453865",

"transactions": {

"amount": {

"currency": "USD",

"total": "1.00"

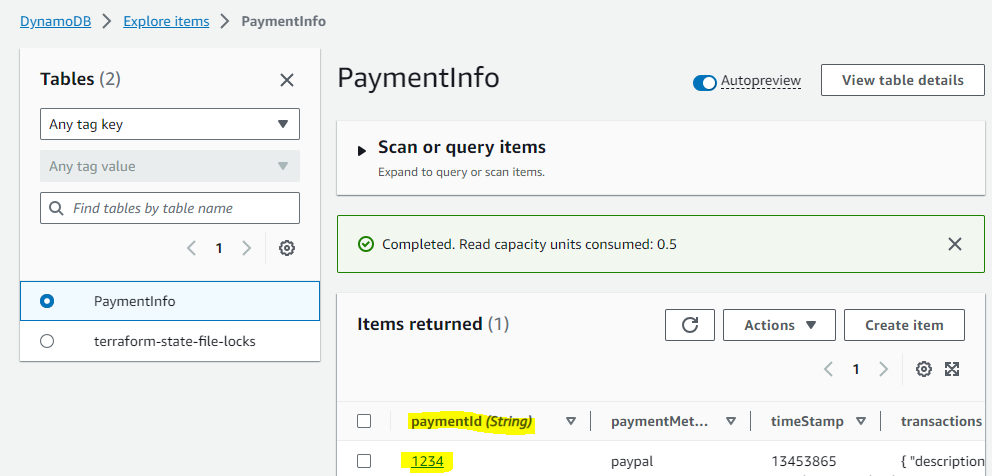
},

"description": "This is the payment description."

}

}

1. The serverless application (Lambda function) receives message from SNS and push the same to SQS.
2. SQS Queue pushes message to another lambda function which validates message and saves in DynamoDB

****