AWS Serverless Project: End-to-End with Canary Deployment

PART 1: Infrastructure Setup

1️⃣ Create DynamoDB Table

- Go to **AWS Console → DynamoDB → Create Table**
- Table name: `ItemsTable`
- Partition key: `id` (String)
- Billing mode: `Pay-per-request`
- Click **Create Table**

PART 2: Initial Lambda Function

2️⃣ Create Lambda Function (v1 – GET & POST)

1. Go to **Lambda → Create Function**
2. Author from scratch:
    - Name: `ItemFunction`
    - Runtime: Python 3.11
    - Permissions: Create new role with basic Lambda permissions

3. Replace code with this:

```
import json
import boto3

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('ItemsTable')

def lambda_handler(event, context):
    method = event['requestContext']['http']['method']

    if method == 'GET':
        item_id = event['queryStringParameters'].get('id')
        response = table.get_item(Key={'id': item_id})
        return {
            'statusCode': 200,
            'body': json.dumps(response.get('Item', {}))
        }

    elif method == 'POST':
        body = json.loads(event['body'])
        table.put_item(Item=body)
        return {
            'statusCode': 201,
            'body': json.dumps({"message": "Item created", "item":
body})
        }

    return {
        'statusCode': 400,
        'body': json.dumps({"error": "Unsupported method"})
    }
```

4. Click **Deploy**

3️⃣ Give Lambda Permission to Access DynamoDB

1. Go to **Lambda → Configuration → Permissions → Execution role**
2. Click the role name
3. Add policy: `AmazonDynamoDBFullAccess`

PART 3: Set Up API Gateway

4️⃣ Create HTTP API

1. Go to **API Gateway → Create API → HTTP API**
2. Add integration:
    - Choose Lambda
    - Select `ItemFunction`
3. Create route:
    - `ANY /items`
4. Create stage: `prod`
5. Click **Deploy**
6. Note the **API Invoke URL**

5️⃣ Test Initial API (GET & POST)

POST:

```
curl -X POST
https://your-api-id.execute-api.region.amazonaws.com/items \
```

```
  -H "Content-Type: application/json" \
  -d '{"id": "123", "name": "Test Item"}'
```

GET:

```
curl
"https://your-api-id.execute-api.region.amazonaws.com/items?id=123"
```

PART 4: Versioning and Advanced Features

6️⃣ Publish Version 1

1. Go to **Lambda → Versions tab**
2. Click **Publish new version**
3. Description: `v1 - GET and POST`

7️⃣ Update Lambda Function (v2 – Add PUT & DELETE)

Edit function code:

```
# Add to existing code
    elif method == 'PUT':
        body = json.loads(event['body'])
        table.put_item(Item=body)
        return {
            'statusCode': 200,
            'body': json.dumps({"message": "Item updated", "item":
body})
        }

    elif method == 'DELETE':
        item_id = event['queryStringParameters'].get('id')
```

```
        table.delete_item(Key={'id': item_id})
        return {
            'statusCode': 200,
            'body': json.dumps({"message": f"Item {item_id} deleted"})
        }
```

Click **Deploy**

8️⃣ Publish Version 2

1. Go to **Versions tab → Publish new version**
2. Description: `v2 - Added PUT and DELETE`

9️⃣ Create Lambda Alias

1. Go to **Aliases tab → Create alias**
    - Name: `prod`
    - Primary version: `1`
2. Under **Additional version weights**, add:
    - Version: `2`
    - Weight: `10%`
3. Click **Create alias**

Now, `prod` splits traffic 90% v1, 10% v2.

PART 5: Hook Alias to API Gateway

🔟 Update API Gateway to Use Alias

1. Go to **API Gateway → Your API → Routes → /items**
2. Click **Edit Integration**
3. Update Lambda ARN to:

```
arn:aws:lambda:region:account-id:function:ItemFunction:prod
```

4. Save integration and redeploy stage

✅ Now API Gateway sends traffic to `ItemFunction:prod`, which routes 90/10 to v1/v2.

PART 6: Test Canary Deployment

Send multiple requests:

- `POST` and `GET` should always work
- `PUT` and `DELETE` may work ~10% of the time

PUT:

```
curl -X PUT https://your-api-id.execute-api.amazonaws.com/items \
  -H "Content-Type: application/json" \
  -d '{"id": "123", "name": "Updated"}'
```

DELETE:

```
curl -X DELETE
"https://your-api-id.execute-api.amazonaws.com/items?id=123"
```