POWER OF AWS CLI

The full form of AWS CLI is **AWS Command Line Interface**. It is a unified tool to manage all the AWS services.

There are lots of options in the AWS console. Using this tool, you don't need to remember or search anything in the AWS console.

Sometimes DevOps engineers need to create the same type of architecture multiple times. In that case, they used to create templates to use those templates multiple times. Those templates are nothing but CLI commands.

I want to show a real-life hands-on example here.

Let's see the **architecture diagram** first.

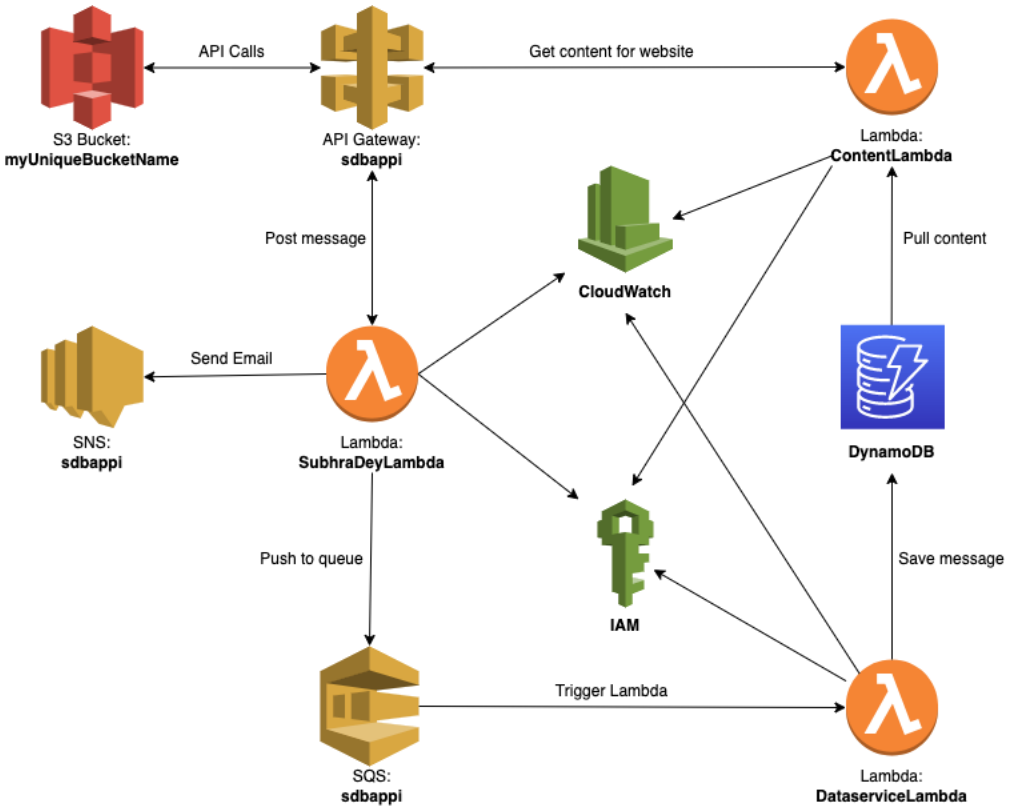

Figure: Architecture diagram for a personal website

PREREQUISITES:

AWS account and an IAM user.

Please check this link to set up an AWS account and create an IAM user:
https://docs.aws.amazon.com/rekognition/latest/dg/setting-up.html

AWS CLI

Please check this link to Install or upgrade and then configure the AWS CLI:
https://docs.aws.amazon.com/codedeploy/latest/userguide/getting-started-configure-cli.html

GETTING STARTED

STEP 1: CREATE THE EXECUTION ROLE

aws iam create-role --role-name sdbappi --assume-role-policy-document '{"Version": "2012-10-17","Statement": [{ "Effect": "Allow", "Principal": {"Service": "lambda.amazonaws.com"}, "Action": "sts:AssumeRole"}]}'

STEP 2: CREATE THE LAMBDA FUNCTIONS

aws lambda create-function --function-name SubhraDeyLambda --runtime python3.9 --zip-file fileb://SubhraDeyLambda.zip --handler lambda_function.lambda_handler --role arn:aws:iam::<AWS Account ID>:role/sdbappi --region us-east-1

aws lambda create-function --function-name DataserviceLambda --runtime python3.9 --zip-file fileb://DataserviceLambda.zip --handler lambda_function.lambda_handler --role arn:aws:iam::<AWS Account ID>:role/sdbappi --region us-east-1

aws lambda create-function --function-name ContentLambda --runtime python3.9 --zip-file fileb://ContentLambda.zip --handler lambda_function.lambda_handler --role arn:aws:iam::<AWS Account ID>:role/sdbappi --region us-east-1

STEP 3: CREATE THE API GATEWAY

aws apigateway create-rest-api --name 'sdbappi' --region us-east-1

STEP 4: CREATE THE TABLES ON DYNAMODB

aws dynamodb create-table --table-name sdbappiEmailCollection --attribute-definitions AttributeName=subject,AttributeType=S --key-schema AttributeName=subject,KeyType=HASH --provisioned-throughput ReadCapacityUnits=2,WriteCapacityUnits=2

aws dynamodb create-table --table-name sdbappiContentCollection --attribute-definitions AttributeName=name,AttributeType=S --key-schema AttributeName=name,KeyType=HASH --provisioned-throughput ReadCapacityUnits=2,WriteCapacityUnits=2

Step 5: Create an SNS topic

aws sns create-topic –name sdbappi

Step 6: Create a queue

aws sqs create-queue –queue-name sdbappi

Step 7: Create a S3 bucket

aws s3api create-bucket –bucket <Bucket Name> –region us-east-1

Step 8: Do the other settings

Do the other settings using AWS CLI. Or you can use the AWS console for further work.

Step 9: Host the application on S3 Bucket

Connect all the APIs to your application and host it to S3 Bucket.

Thank you
Author: Subhra Dey
Date: Nov 23, 2022