API Gateway - Lambda - EFS - S3

23 DEC 2021 Hai Tran

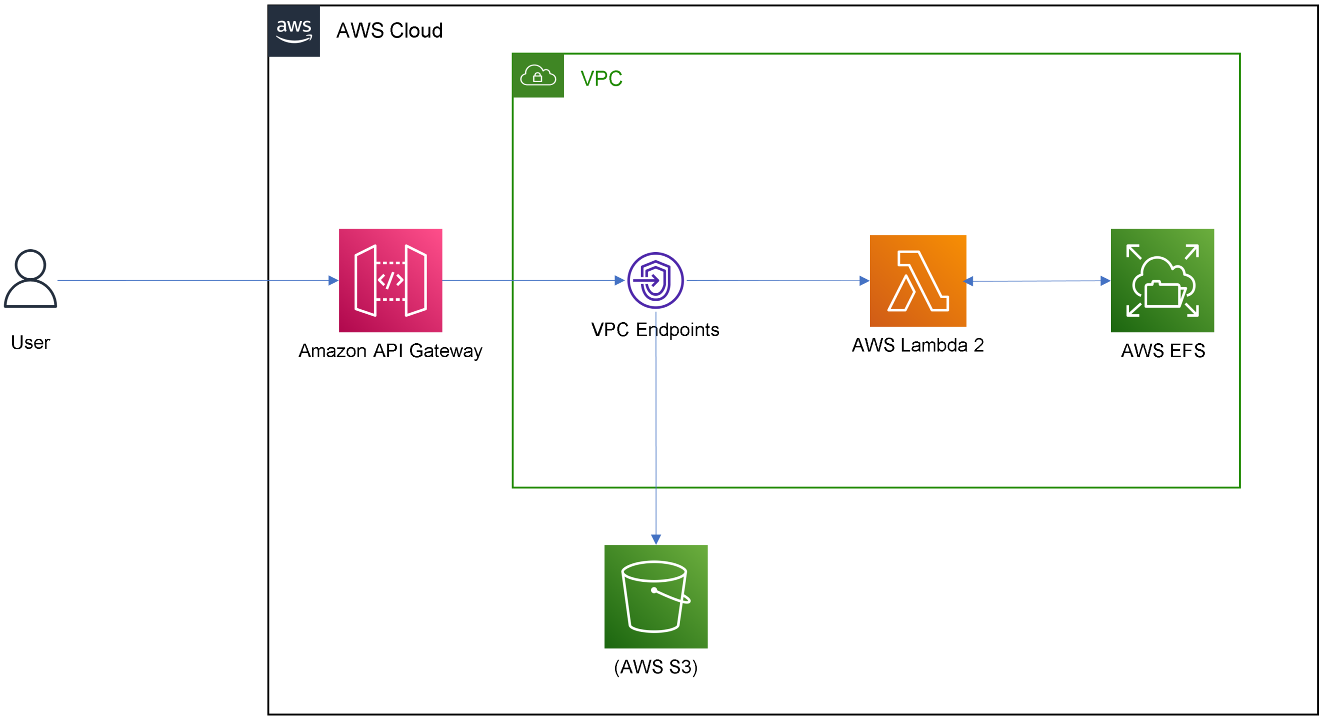
Discussion

I want to create a API endpoint to process data uploaded from users and stored in a S3 bucket. Here is something I did but not sure is this a good practice or not. So I would like to listen to comments.

- Lambda depdenccies is several GB so not able to zip and deploy, so I load dependencies from EFS reference
- Since the Lambda is inside a VPC, it is not able to access S3. So I add a VPC S3 endpoint.

Architecture

I took Lambda over EC2 and ECS because the processing tasks requires 10GB RAM, 20 seconds, need to scale out, and the load is not always on. Also Lambda has no IP and port, process data without traverse the internet, easy to operate, and monitor. It can scale out by using a pool of multiple such as 10 Lambdas and keep warm, I don't like provisioned because the load/spike is unpredictable. However, this wwill needs some code to route/schedule requests (like a simple roud robin) to Lambdas. Is a simple round robin good and robust enough?



Step 1. Create a python handler just to check

```
def lambda_handler(event, context):
    # parse parameters
    params = event["queryStringParameters"]["filename"]
    #
    return {
        'statusCode': 200,
        'body': json.dumps({'filename': params})
    }
```

Step 2. Check CORS for Lambda and curl

Need to add HEADER to the lambda handler

```
def lambda_handler(event, context):
    # parse parameters
    filename = event["queryStringParameters"]["filename"]
    # client s3 client to read an object from s3
    s3Client = boto3.client("s3")
    items = s3Client.list_objects(
        Bucket='bucketname'
    )
    # response
    return {
        'statusCode': 200,
        'headers': {
            "Access-Control-Allow-Origin": "*",
            "Access-Control-Allow-Headers": "Content-Type",
            "Access-Control-Allow-Methods": "OPTIONS,GET"
        },
        'body': json.dumps({'filename': filename, 'item': items},
indent=4, sort_keys=True, default=str)
    }
```

Step 3. Configure Lambda access a S3 bucket

- Need a VPC endpoint.
- IAM role for the Lambda function to access S3 reference

```


  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ExampleStmt",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET/*"
      ]
    }
  ]
}
```

Step 4. Create a EFS endpoint

- create a EFS file system Reference
- create an access point Reference
- CloudFormation template

```
  AccessPointResource:
    Type: 'AWS::EFS::AccessPoint'
    Properties:
      FileSystemId: !Ref FileSystemResource
      PosixUser:
        Uid: "1000"
        Gid: "1000"
      RootDirectory:
        CreationInfo:
          OwnerGid: "1000"
          OwnerUid: "1000"
          Permissions: "0777"
        Path: "/efs"
```

Step 5. Mount EFS access point to EC2

- Install efs-utils for ubuntun git repo
- mount the access point on an EC2 mount
- mount access point command
- ensure that EC2 and EFS is in the default security group

```
mount -t efs -o tls,accesspoint=fsap-12345678 fs-12345678: /localmountpoint
```

Step 6. Add polcies for Lambda

- add VPC endpoint S3 because now this is the only way for Lambda to access S3
- policy to add the VPC
- policy to read and write the EFS
- policy to read the S3 bucket (just double check)
- test

```
import json
import boto3

def lambda_handler(event, context):
    # test read write EFS
    with open("/mnt/efs/data.txt", "w") as file:
        file.write("Hello from Lambda")
    # test lamba access s3
    s3Client = boto3.client("s3")
    items = s3Client.list_objects(
        Bucket='haitran-swinburne-2021'
    )
    # parse parameters
    filename = event["queryStringParameters"]["filename"]
    # response
    return {
        'statusCode': 200,
        'headers': {
            "Access-Control-Allow-Origin": "*",
            "Access-Control-Allow-Headers": "Content-Type",
            "Access-Control-Allow-Methods": "OPTIONS,GET"
        },
         'body': json.dumps({'filename': filename, 'item': items},
indent=4, sort_keys=True, default=str)
    }
```

Step 7. Configure Lambda load dependencies from EFS

- recommendation and performance from AWS

Reference

1. Choosing Your VPC Endpoint Strategy for Amazon S3
2. Lambda EFS Deep Learning
3. AWS S3 access with VPC endpoints
4. aws s3 vpc endpoint
5. three ways to use AWS services from