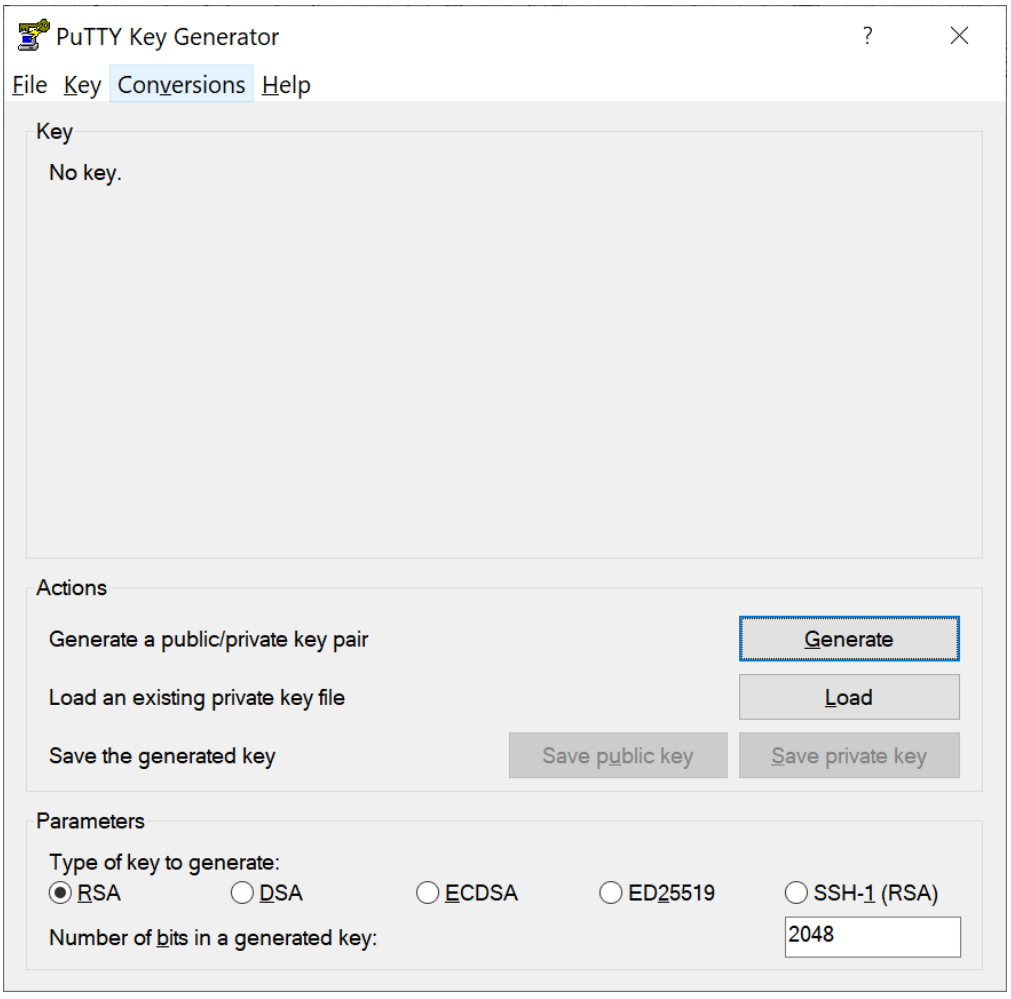
Capgemini challenge 2021: Create a wallet service
James Thomas DSouza

The below steps show how to deploy the wallet service

PART 1: SETUP YOUR MACHINE IN SINGAPORE REGION

STEP 1: Create a t2.large EC2 machine. Make sure Security Group is configured to allow SSH port 22 from your IP address. EC2 has a public IP address assigned. Download the privatekey.pem file
STEP 2: Install putty on your local box
STEP 3: Start puttygen on your local box. In menu, click Conversions and import key. Choose privatekey.pem



STEP 5: Click Save Private key and save the file. It will be saved as in ppk extension. Example privatekey.ppk

PART 2: INSTALL GIT on your local box using the below commands

sudo yum install git -y

```
mkdir mygit
```

```
cd mygit/
```

```
git init
```

```
ls -a
```

```
ls -a .git/
```

PART 3: DOWNLOAD THE SOURCE CODE TO INSTALL WALLET SERVICE

git clone https://github.com/aws-samples/serverless-wallet

PART 4: get inside serverless_wallet folder and rename config_file.py_sample to

config_file.py

mv config_file.py_sample config_file.py

PART 5: CONFIGURE AWS

aws configure

PART 6: Install npm and nodejs

5. Install npm package manager by installing nodejs using below commands

Command 1:

```
curl -o-

https://raw.githubusercontent.com/creationix/nvm/v0.32.0/install.sh |

bash
```

Command 2:

```
. ~/.nvm/nvm.sh
```

Command 3:

```
nvm install 16.4.2
```

Command 4:

```
node -e "console.log('Running Node.js ' + process.version)"
```

Command 5:

```
npm
```

PART 7: INSTALL AWS CDK

```
npm install -g aws-cdk
```

```
PART 8: Install Python 3.8
```

11. How to install python 3.8

sudo yum install gcc openssl-devel bzip2-devel libffi-devel

cd /opt

sudo wget https://www.python.org/ftp/python/3.8.7/Python-3.8.7.tgz

sudo tar xzf Python-3.8.7.tgz

cd Python-3.8.7

sudo ./configure --enable-optimizations

sudo make altinstall

python3.8 -V

PART 9: Set python3 = python 3.8

```
sudo chmod +rwx /usr/bin/python3
```

```
sudo ln -sf /usr/local/bin/python3.8 /usr/bin/python3
```

PART 10: Install Docker

```
sudo yum update -y

sudo amazon-linux-extras install docker

sudo yum install docker

sudo service docker start

sudo usermod -a -G docker ec2-user

sudo usermod -aG docker ${USER}
```

PART 11: open src folder, Install Python libraries of CDK

```
pip3 install -r requirements.txt
```

PART 12: Close your putty session and relogin into EC2. Run the below commands

```
nvm install 16.4.2

sudo systemctl start docker

sudo docker run hello-world

docker info
```

PART 12: Update settings in config_file.py

change directory to src folder inside serverless-wallet

update account number and region inside config_file.py using vi editor

update ‘log_level’ = ‘DEBUG’

To insert press esc key followed by ‘I’ key on keyboard

To save press esc key followed by ‘:wq’ keys on keyboard

PART 13: Deploy your architecture by running below two commands

```
cdk bootstrap
```

```
cdk deploy
```

PART 14: Verify if everything is installed

Close your putty session

Open AWS Management console.

See Lambda services. See if you can see your wallet functions

See API Gateway. See if you can see your API functions

Part 15: Open QLDB

Verify if there is a ledger ‘wallet’

PART 16:

Open Query editor and run the below two queries

```
CREATE TABLE Wallet

CREATE INDEX ON Wallet (accountId)
```

PART 17:How to test your Lambda functions

Go to management console and open Lambda function createaccount

Open Test tab , create a event test, and put the below code and click Test

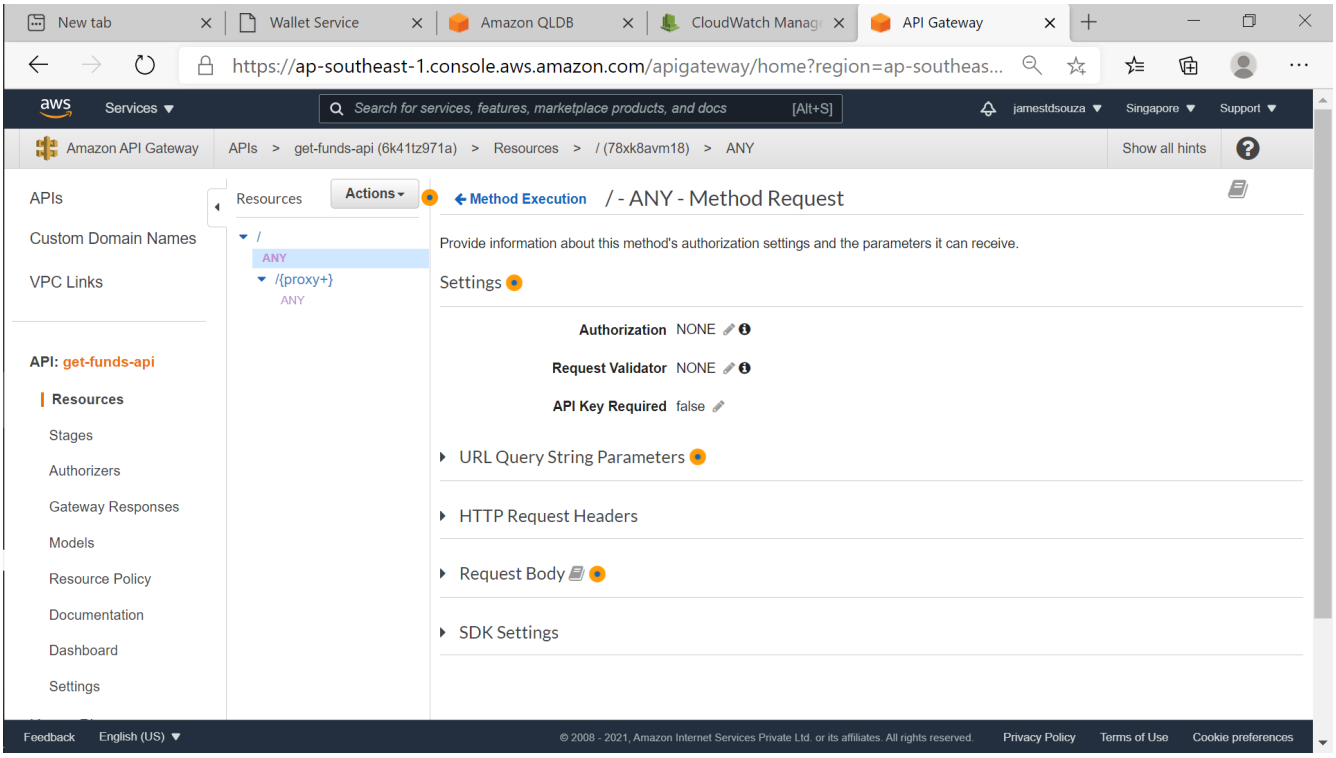
```
{
  "httpMethod": "POST",
  "Authorization": null,
  "queryStringParameters": null,
  "pathParameters": null,
  "body": "{\"accountId\":\"1323\"}"
}
```

You should see a record created in QLDB with accountId = 1323. In QLDB you can verify by running the “SELECT * FROM Wallet”

queryStringParameters

PART 18:How to test your API function

Before you start using API, it is recommended for testing purpose you set Authentication to NONE for every API



In management console, go to API, Open API method createaccount and click Test

select method=POST

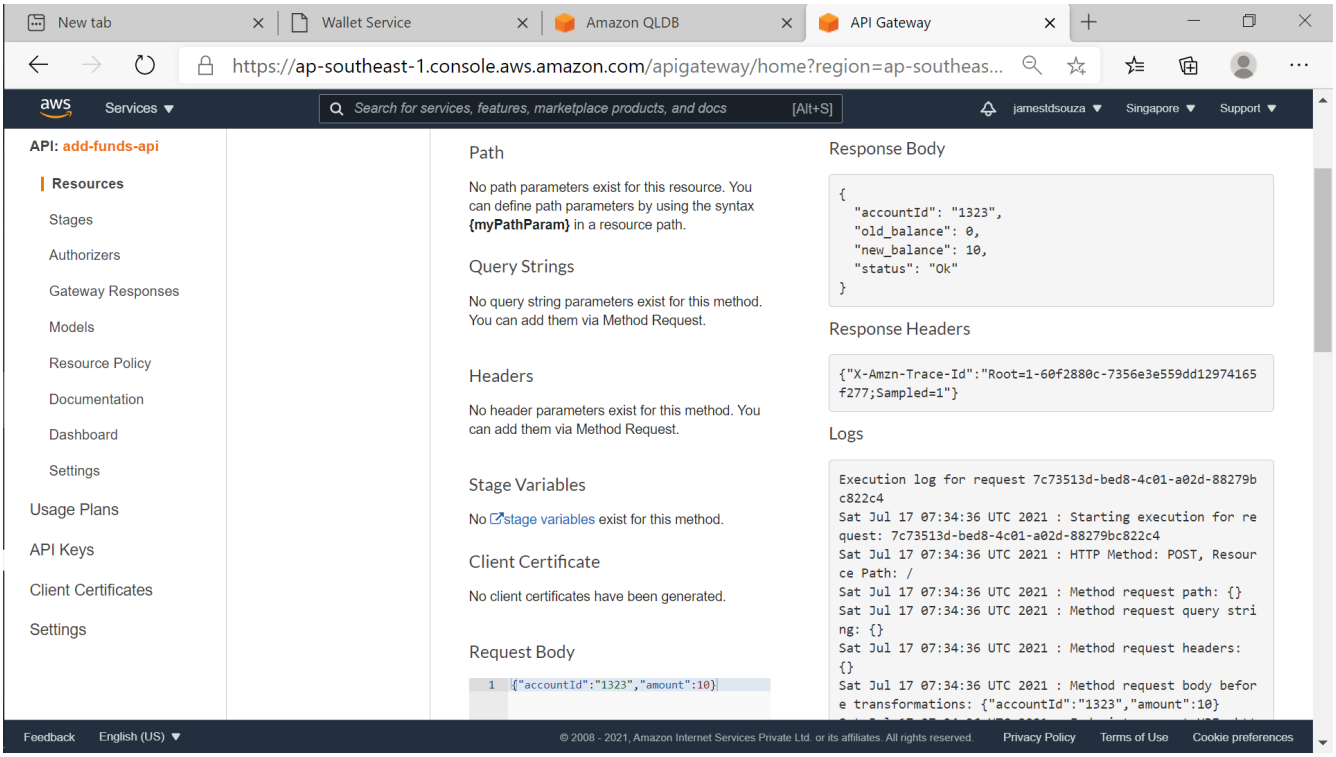
paste the below in body section and click Test.

```
{"accountId":"1323"}
```

for adding funds, use the below

```
{"accountId":"1323","amount":10}
```

the output will look as follows:

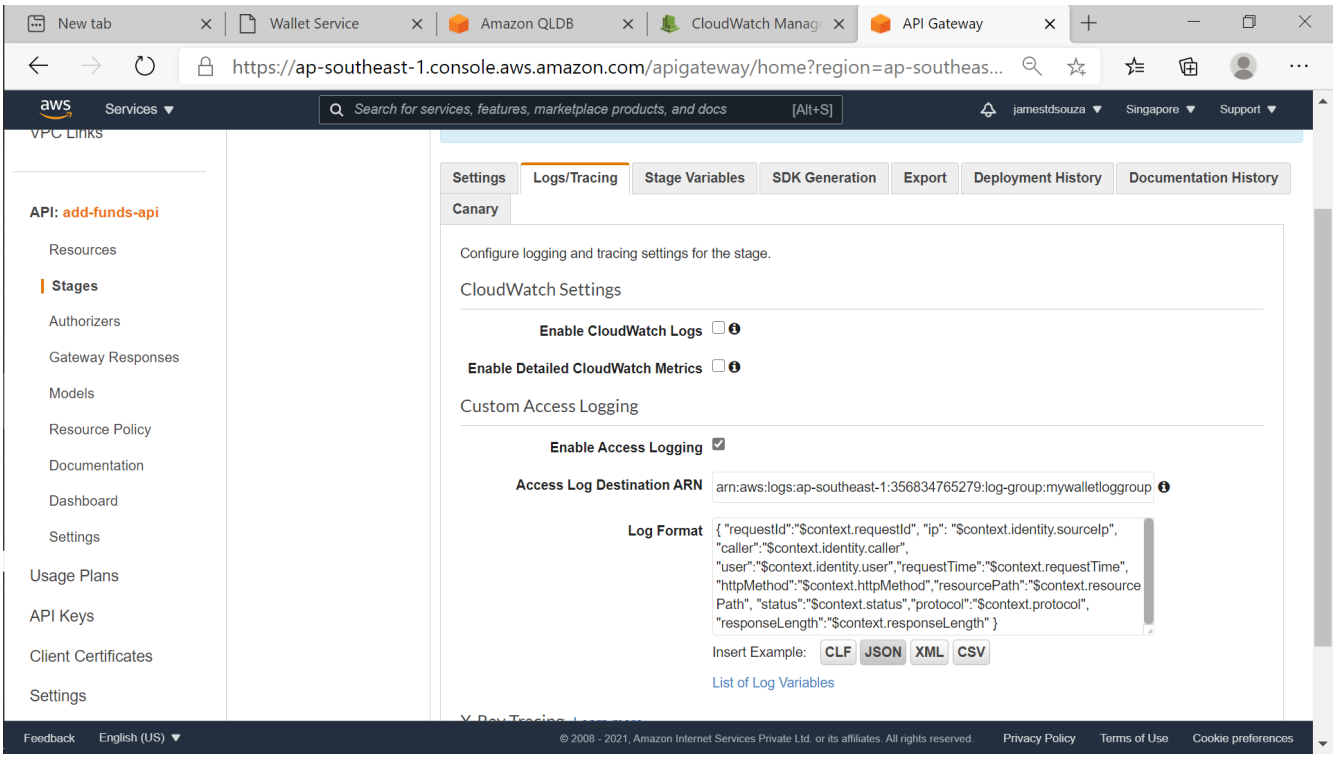


In case of createAccount, An account will be created successfully if it does not exists. You can verify the same in QLDB

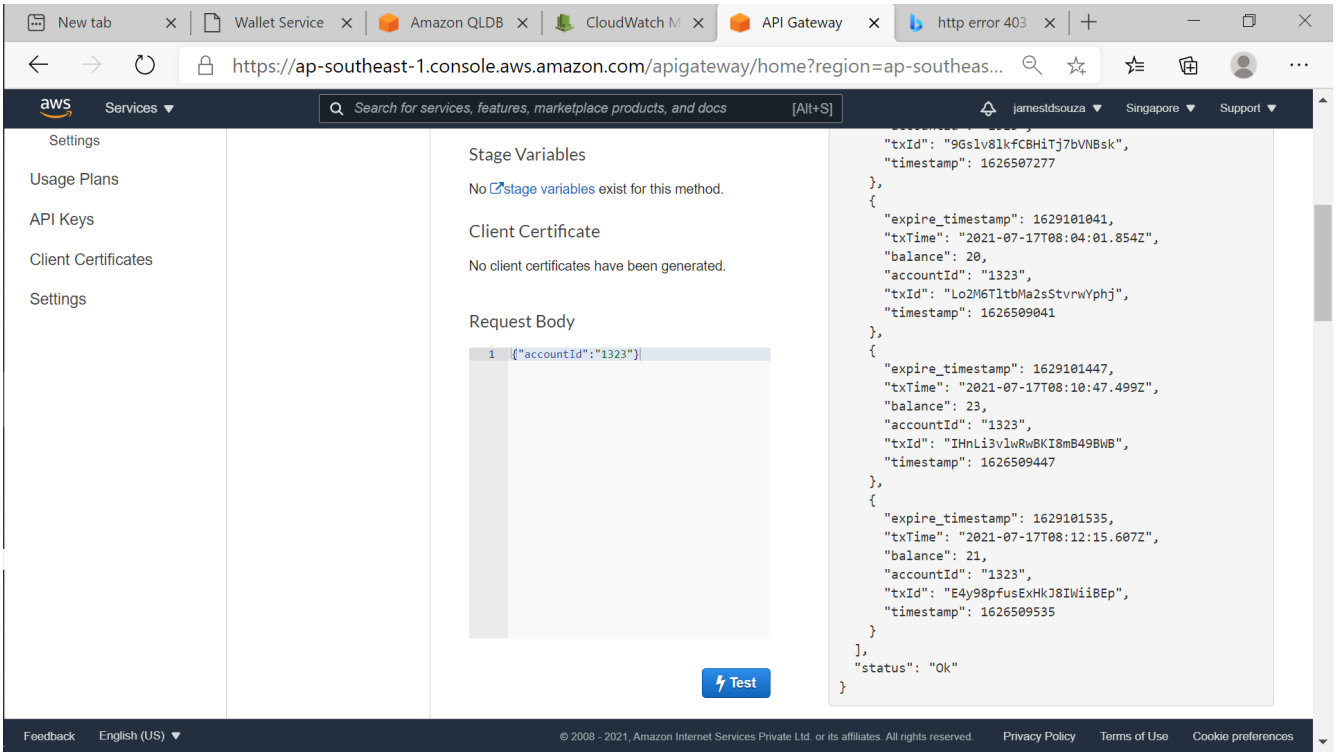
PART 19: How to set up Cloudwatch to track API calls

It is recommended that you create a log group “mywalletloggroup” in CloudWatch to track API calls.

In API gateway, for each and every API that you create, it is recommended that you enable access logs as seen below.



Note Get Funds API returns balance and Get Transaction History returns the chain of transactions that happened



PART 19: It is recommended that you create a front end web site or HTML page to call all API function