**Building wallet service from scratch: wallet-app**

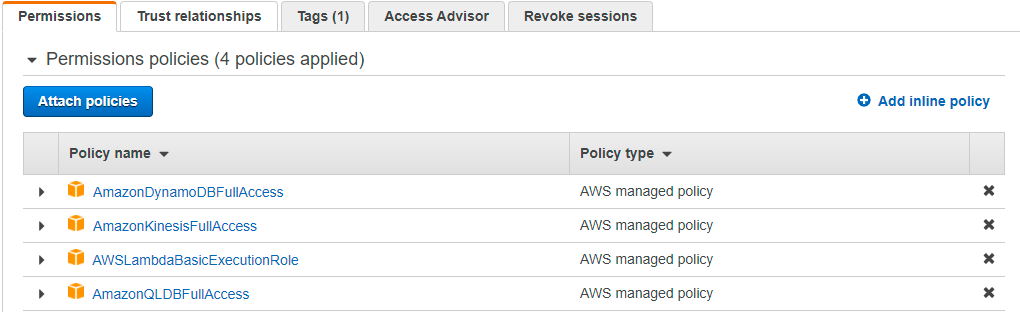
Overall build and deployment in divided into multiple steps. Follow below steps 1 by 1 to create this service application in your AWS account. Some important points to note before getting started.

**Important points:**

1. AWS account should be created upfront.
2. AWS account used for building this service is root account. (What does it mean? – I didn’t create any user, but used default AWS root account)
3. Region used – **us-east-1.**
4. Approach for building application is – Serverless Application (SAM)
5. Python 3.8 is used as language for app.
6. Below local installations required before starting development-
7. Python 3.8
8. aws-sam-cli
9. aws-cli (v2)
10. docker
11. VS Code (preferred IDE)

**Now – Steps**

1. Log in AWS account
2. You need to create – **Roles, QLDB ledger, Table, Index and QLDB stream from console**. All other resources will be taken care by SAM package. (Serverless Application)
3. Below are specifications (\*Follow names properly, they will be used in SAM template):
   1. \*Role 1: wallet-app-lambda
      1. Name: **wallet-app-lambda**
      2. Managed policies: Attach below managed policies to role



* 1. \*Role 2: **wallet-stream-records**
     1. Name: **wallet-stream-records**
     2. Managed Policy: **QLDBStreamKinesisPermissions**
  2. \*QLDB Ledger: **wallet-ledger**
     1. Name: **wallet-ledger**
     2. Permission Mode: **Standard**
  3. \*QLDB Table: **user\_account**
     1. Name: **user\_account**
  4. \*\*QLDB Stream: **wallet-records-stream**
     1. Name: **wallet-records-stream**
     2. Kinesis Stream: Choose Kinesis stream created by SAM package stack.

\*Keep Names uniform as mentioned to avoid any issues.

\*\*Create QLDB stream after you build and deploy SAM stack package. As it is dependent on Kinesis Stream created by SAM package

1. Now, extract zip file at your project directory in local system
   1. Code is also available at GitHub at -
2. Make sure you have docker running on your system and all other requirements fulfilled as mentioned in section ‘Important points’.
3. Setup AWS credentials by using command “**aws configure**” and pass your accessKey and secreteKey as per instructions. (You can get your accessKey and secretKey for AWS console)
4. Run “**sam build –use-container**” from cmd/powershell or VS Code.
5. After build completed successfully. Run “**sam deploy**”. For changeset related question put **Y** as answer
6. After successful deployment, your application will be available on AWS account.
7. Now setup QLDB Steam with specification mentioned in **step 3**. Make sure choosing kinesis stream we created by package while creating QLDB Steams.
8. Now, go to AWS console>API Gateway>select wallet-app (app name)> stage (left side)> you can access your **api-urls for your all services.**
9. You can enable cache for API by setting **TTL** in API gateway console. It costs on Hourly basis; So, I have not included same in my SAM template.
10. Your wallet-app (app name) service is up and running now.

**Instructions for testing services:**

1. Use below request body for REST API –

{

"userAccountId":"test123",

"Amount":5,

"Name":"abc",

"Balance":100

}

1. Here, Amount is used when you want **to add or withdraw** funds to/from account.
2. **userAccountId, Name and Balance** are **must** keys for **createUser** service.

Below are APIs build by me. These will be made available till 30th July and I will close after it as it may cost high.

**# API 1 - POST**

https://ncy7xi9hz2.execute-api.us-east-1.amazonaws.com/Prod/createUser

**# API 2- GET**

https://ncy7xi9hz2.execute-api.us-east-1.amazonaws.com/Prod/getFunds/abc123

**# API 3 - PUT**

https://ncy7xi9hz2.execute-api.us-east-1.amazonaws.com/Prod/addFunds/1

**# API 4 - PUT**

https://ncy7xi9hz2.execute-api.us-east-1.amazonaws.com/Prod/withdrawFunds/1

**#API 5 – GET – Historical User Account Change**

https://ncy7xi9hz2.execute-api.us-east-1.amazonaws.com/Prod/getRecords/test1245

{

"userAccountId":"test123",

"Amount":5,

"Name":"abc",

"Balance":100

}