Running Notes:

New controller code for feign client

AccountsController.java

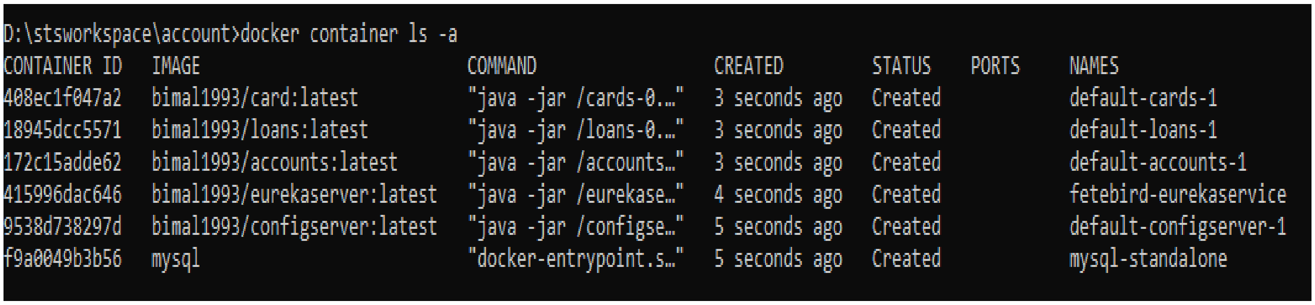
```
package com.bank.account.controller;

@RestController
public class AccountsController {
@Autowired
private AccountsRepository accountRepository;
@Autowired
LoansFeignClient loansFeignClient;
@Autowired
CardsFeignClient cardsFeignClient;
@Autowired
AccountsServiceConfig accountsConfig;

@PostMapping("/myaccount")
public Accounts getAccountDetails(@RequestBody Customer customer) {
Accounts account = accountRepository.findByCustomerId(customer.getCustomerId());
if (account != null) {
return account;
} else {
return null;
}
}

@GetMapping("/account/properties")
public String getPropertyDetails() throws JsonProcessingException {
ObjectWriter ow = new ObjectMapper().writer().withDefaultPrettyPrinter();
Properties properties = new Properties(accountsConfig.getMsg(), accountsConfig.getBuildVersion(),
accountsConfig.getMailDetails(), accountsConfig.getActiveBranches());
String jsonStr = ow.writeValueAsString(properties);
return jsonStr;
}

@PostMapping("/myCustomerDetails")
public CustomerDetails myCustomerDetails(@RequestBody Customer customer) {
Accounts accounts = accountRepository.findByCustomerId(customer.getCustomerId());
List<Loans> loans = loansFeignClient.getLoansDetails(customer);
List<Cards> cards = cardsFeignClient.getCardDetails(customer);

CustomerDetails customerDetails = new CustomerDetails();
customerDetails.setAccounts(accounts);
customerDetails.setLoans(loans);
customerDetails.setCards(cards);

return customerDetails;

}
```

To build docker images:

- docker build . -t bimal1993/accounts
- docker build . -t bimal1993/loans
- docker build . -t bimal1993/card
- docker build . -t bimal1993/configserver
- docker build . -t bimal1993/eurekaserver

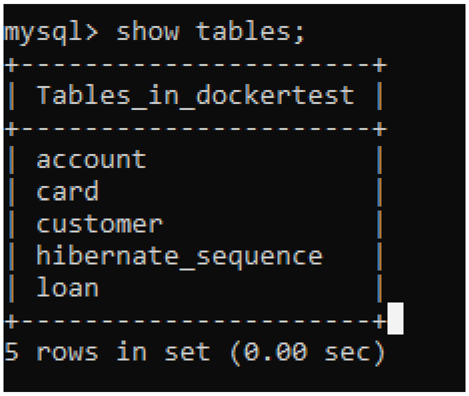
```
docker run --network spring-net -p 8080:8081 bimal1993/accounts
docker run --network spring-net -p 8090:8090 bimal1993/loans
docker run --network spring-net -p 9000:9000 bimal1993/card
```

D:\stsworkspace\account\docker-compose\default>docker compose up



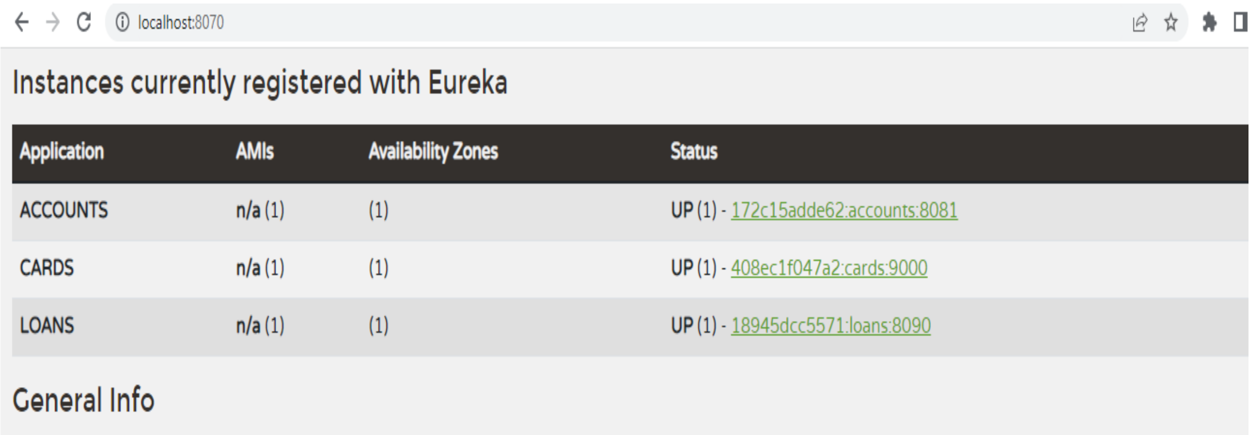
```
D:\stsworkspace\account>docker container ls -a
CONTAINER ID   IMAGE                           COMMAND                  CREATED         STATUS    PORTS     NAMES
408ec1f047a2   bimal1993/card:latest           "java -jar /cards-0.…"   3 seconds ago   Created             default-cards-1
18945dcc5571   bimal1993/loans:latest          "java -jar /loans-0.…"   3 seconds ago   Created             default-loans-1
172c15adde62   bimal1993/accounts:latest       "java -jar /accounts…"   3 seconds ago   Created             default-accounts-1
415996dac646   bimal1993/eurekaserver:latest   "java -jar /eurekase…"   4 seconds ago   Created             fetebird-eurekaservice
9538d738297d   bimal1993/configserver:latest   "java -jar /configse…"   5 seconds ago   Created             default-configserver-1
f9a0049b3b56   mysql                           "docker-entrypoint.s…"   5 seconds ago   Created             mysql-standalone
```

docker container exec -it be18f2a16232 bash
mysql -uroot -p
1234



```
mysql> show tables;
+----------------------+
| Tables_in_dockertest |
+----------------------+
| account              |
| card                 |
| customer             |
| hibernate_sequence   |
| loan                 |
+----------------------+
5 rows in set (0.00 sec)
```

To check docker log
D:\stsworkspace\loans>docker logs -f e0f509001613



localhost:8070

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
ACCOUNTS	n/a (1)	(1)	UP (1) - 172c15adde62:accounts:8081
CARDS	n/a (1)	(1)	UP (1) - 408ec1f047a2:cards:9000
LOANS	n/a (1)	(1)	UP (1) - 18945dcc5571:loans:8090

General Info

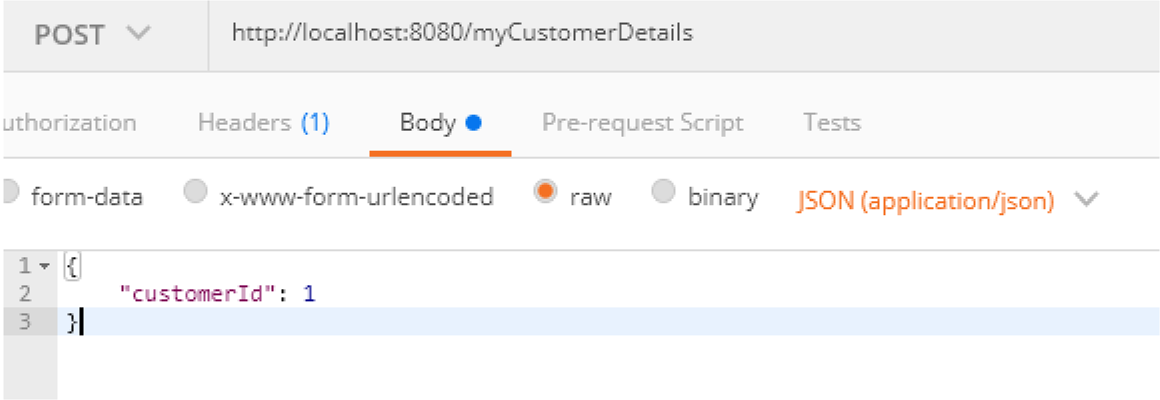
docker-compose.yml

```
version: "3.8"

services:

  mysql-standalone:
    image: mysql
    container_name: mysql-standalone
    networks:
    - spring-net
    environment:
      - MYSQL_ROOT_PASSWORD=1234
      - MYSQL_DATABASE=dockertest

  configserver:
    image: bimal1993/configserver:latest
    mem_limit: 700m
    ports:
      - "8071:8071"
    networks:
     - spring-net

  eurekaserver:
    image: bimal1993/eurekaserver:latest
    container_name: fetebird-eurekaservice
    mem_limit: 700m
    ports:
      - "8070:8070"
    networks:
     - spring-net
    depends_on:
      - configserver
    deploy:
      restart_policy:
        condition: on-failure
        delay: 25s
        max_attempts: 10
        window: 120s
    environment:
      SPRING_PROFILES_ACTIVE: default
```

```yaml
      SPRING_CONFIG_IMPORT: configserver:http://configserver:8071/

  accounts:
    image: bimal1993/accounts:latest
    mem_limit: 700m
    ports:
      - "8080:8081"
    networks:
      - spring-net
    depends_on:
      - eurekaserver
      - mysql-standalone
    deploy:
      restart_policy:
        condition: on-failure
        delay: 60s
        max_attempts: 10
        window: 120s
    environment:
      SPRING_PROFILES_ACTIVE: default
      SPRING_CONFIG_IMPORT: configserver:http://configserver:8071/

  loans:
    image: bimal1993/loans:latest
    mem_limit: 700m
    ports:
      - "8090:8090"
    networks:
      - spring-net
    depends_on:
      - eurekaserver
      - mysql-standalone
    deploy:
      restart_policy:
        condition: on-failure
        delay: 60s
        max_attempts: 10
        window: 120s
    environment:
      SPRING_PROFILES_ACTIVE: default
      SPRING_CONFIG_IMPORT: configserver:http://configserver:8071/

  cards:
    image: bimal1993/card:latest
    mem_limit: 700m
    ports:
      - "9000:9000"
    networks:
      - spring-net
    depends_on:
      - eurekaserver
      - mysql-standalone
    deploy:
      restart_policy:
```

```
    condition: on-failure
    delay: 60s
    max_attempts: 10
    window: 120s
  environment:
   SPRING_PROFILES_ACTIVE: default
   SPRING_CONFIG_IMPORT: configserver:http://configserver:8071/

networks:
 spring-net:
```

Request:



POST http://localhost:8080/myCustomerDetails

● raw JSON (application/json)

```
{
    "customerId": 1
}
```

```
{
  "accounts": {
    "customerId": 1,
    "accountNumber": 186576453,
    "accountType": "Savings",
    "branchAddress": "123 Main Street, New York",
    "createDt": "2022-07-09"
  },
  "loans": [
    {
      "loanNumber": 3,
      "customerId": 1,
      "startDt": "2021-02-14",
      "loanType": "Home",
      "totalLoan": 50000,
      "amountPaid": 10000,
      "outstandingAmount": 40000,
      "createDt": "2018-02-14"
    },
    {
      "loanNumber": 1,
      "customerId": 1,
      "startDt": "2020-10-13",
      "loanType": "Home",
        "totalLoan": 200000,
        "amountPaid": 50000,
        "outstandingAmount": 150000,
        "createDt": "2020-10-13"
      },
      {
        "loanNumber": 2,
        "customerId": 1,
        "startDt": "2020-06-06",
        "loanType": "Vehicle",
        "totalLoan": 40000,
        "amountPaid": 10000,
        "outstandingAmount": 30000,
        "createDt": "2020-06-06"
      },
      {
        "loanNumber": 4,
        "customerId": 1,
        "startDt": "2018-02-14",
        "loanType": "Personal",
        "totalLoan": 10000,
        "amountPaid": 3500,
        "outstandingAmount": 6500,
        "createDt": "2018-02-14"
      }
    ],
    "cards": [
      {
        "cardId": 1,
        "customerId": 1,
        "cardNumber": "4565XXXX4656",
        "cardType": "Credit",
        "totalLimit": 10000,
        "amountUsed": 500,
        "availableAmount": 9500,
        "createDt": "2022-07-09"
      },
      {
        "cardId": 2,
        "customerId": 1,
        "cardNumber": "3455XXXX8673",
        "cardType": "Credit",
        "totalLimit": 7500,
        "amountUsed": 600,
        "availableAmount": 6900,
        "createDt": "2022-07-09"
      },
      {
```

```
      "cardId": 3,
      "customerId": 1,
      "cardNumber": "2359XXXX9346",
      "cardType": "Credit",
      "totalLimit": 20000,
      "amountUsed": 4000,
      "availableAmount": 16000,
      "createDt": "2022-07-09"
    }
  ]
}
```

http://localhost:8071/accounts/default

```
{
  "name": "accounts",
  "profiles": [
    "default"
  ],
  "label": null,
  "version": "0484f35409d0c637fdf511424fb30646e6607417",
  "state": null,
  "propertySources": [
    {
      "name": "https://github.com/Bimal1993/microservice-config.git/accounts.properties",
      "source": {
        "accounts.msg": "Welcome to the EazyBank Accounts Default application",
        "accounts.build-version": "3",
        "accounts.mailDetails.hostName": "default-accounts@eazybytes.com",
        "accounts.mailDetails.port": "9000",
        "accounts.mailDetails.from": "default-accounts@eazybytes.com",
        "accounts.mailDetails.subject": "Your Account Details from Eazy Bank Default Environment",
        "accounts.activeBranches[0]": "Mumbai",
        "accounts.activeBranches[1]": "London",
        "accounts.activeBranches[2]": "Washington"
      }
    }
  ]
}
```

http://localhost:9000/cards/properties

```
{
 "msg" : null,
 "buildVersion" : "3",
 "mailDetails" : {
  "hostName" : "default-cards@eazybytes.com",
  "port" : "9002",
  "from" : "default-cards@eazybytes.com",
    "subject" : "Your Card Details from Eazy Bank Default Environment"
  },
  "activeBranches" : [ "Mumbai", "London", "Washington" ]
}
```