ANNEXURE – 4

Oracle Banking Branch

Release 14.6.1.0.0

Part Number F61366-01

August 2022

ORACLE®

Table of Contents

ORACLE®

1. ANNEXURE - 4

1.1 <u>Introduction</u>

This guide is a supporting document for the installation of Oracle Banking Microservices Architecture applications. You can find the reference in the respective installation guides.

1.2 <u>How to do Multi Node setup (High Availability Architecture)</u>

1.2.1 <u>Configuration Server Related Changes</u>

The below changes are to be made in the PROPERTIES table pointed to by the Configuration Server:

- **For the Discovery Server:**

    PLATO Discovery Service should have an entry for its entire peer PLATO Discovery Services configured through **eureka.client.serviceUrl.defaultZone**. This will hold a comma-separated list of all the peer PLATO Discovery services.

    In addition, to enable the peer aware mode for the PLATO Discovery Service we should set the **eureka.client.register-with-eureka** to true.

ID	APPLICATION	PROFILE	LABEL	KEY	VALUE
1	plato-discovery-service	jdbc	jdbc	eureka.client. serviceUrl.de faultZone	http://<IP of the server where the first instance of PLATO Discovery Service is running>:<PORT where the first instance of PLATO Discovery Service is running>/plato-discovery-service/eureka,http://<IP of the server where the second instance of PLATO Discovery Service is running>:<PORT where the second instance of PLATO Discovery Service is running>/plato-discovery-service/eureka
2	plato-discovery-service	jdbc	jdbc	eureka.client. register-with-eureka	true
3	plato-discovery-service	jdbc	jdbc	server.port	<< PORT Number where the PLATO Discovery Service is running >>

ORACLE

- **For the Individual Services:**

  Each service should have an entry of all the PLATO Discovery Services configured through **eureka.client.serviceUrl.defaultZone**. This will hold a comma separated list of all the PLATO Discovery services.

ID	APPLICATION	PROFILE	LABEL	KEY	VALUE
1	<<service-name>>	jdbc	jdbc	eureka.client.serviceUrl.defaultZone	http://<IP of the server where the first instance of PLATO Discovery Service is running>:<PORT where the first instance of PLATO Discovery Service is running>/plato-discovery-service/eureka,http://<IP of the server where the second instance of PLATO Discovery Service is running>:<PORT where the second instance of PLATO Discovery Service is running>/plato-discovery-service/eureka

1.2.2 Plato UI Configuration Server Related Changes

For each of the product registered in **PRODUCT_SERVICES_ENV_LEDGER**, we need to change the URL to point to the Load Balancer of the PLATO API Gateway Service.

ID	PRODUCT_NAME	URL
1	<<PRODUCT NAME>>	<< HTTP URL OF THE LOAD BALANCER >>

1.2.3 setDomainEnv.sh related changes

- **For all the Micro Services:**

  Individual MICRO services should now access the PLATO Config Service via the Load Balancer URI (i.e). configured in the server runtime through the property **plato.services.config.uri**.

  The **plato.services.config.uri** must point to the URI of the load balancer. The format of the same would be as follows:

  ```
  -Dplato.services.config.uri=http://<< IP OF THE LOAD BALANCER
  >>:<< PORT OF THE LOAD BALANCER >>
  ```

- **For the UI APPSHELL:**

  UI APPShell should now access the Gateway Service via the Load balancer URI (i.e) configured in the server runtime. For example, **Dapigateway.url**.

  The **apigateway.url** must point to the host and port of the load balancer. The format of the same would be as follows:

  ```
  -Dapigateway.url=http://<< IP OF THE LOAD BALANCER >>:<< PORT OF
  THE LOAD BALANCER >>
  ```

ORACLE

If you need to install the services of Oracle Banking Microservices Architecture in more than two nodes, it is not possible to maintain the value of the eureka URL in the properties table due to the size restriction. In such cases, you can remove the following key from the properties table and add in the *setuseroverrides.sh* file.

```
-Deureka.client.serviceUrl.defaultZone
```

1.2.4 Requirement of Load Balancers

Load Balancers are required for PLATO API GATEWAY Service, PLATO Configuration Service, and PLATO UI APP SHELL.

1.2.4.1 PLATO API Gateway Service

PLATO API Gateway Service acts as a single point of entry for UI and External Systems to access the underlying services. In a multi node deployment where multiple PLATO API Gateway Services are deployed, we would need a single URI for accessing the multi node deployments of the PLATO API Gateway Services. This Load Balancer would help us to achieve that functionality.

1.2.4.2 PLATO Configuration Service

All the domain services access PLATO Configuration Service for retrieving their configurations. In a multi node deployment where multiple PLATO Configuration Services are deployed, we would need a single URI for accessing the multi node deployments of the PLATO Configuration Services. This Load Balancer would help us to achieve that functionality.

1.2.4.3 PLATO UI APP SHELL

 The PLATO UI App Shell acts as the single user interface entry point for the users. In a multi-node deployment, where multiple instances of PLATO UI APP SHELL are deployed, users need a single URI for accessing the multi-node deployments of the PLATO UI APP SHELL. Load Balancer setup will help to achieve this.

In addition to the "App Shell," the UI of the application is serviced by additional UI "component server" applications. These are for SMS, CMC, MOC, and the respective product domain too. All these UI component server applications need to be deployed in the same managed server, where PLATO UI APP SHELL war is deployed.

If the deployment is in a cluster with more than one managed server for UI applications, then all the UI applications need to be deployed in the clustered managed servers, and appropriate load balancer setup need to be done for all the UI applications.

1.3 NGINX LOAD BALANCER SERVICES

This section contains the following sub-sections:

- Installation and Setup
- Load Balancer Route Configurations

ORACLE®

1.3.1 Installation and Setup

1.3.1.1 Download

1. Download the tar file from **nginx.org/download/** Index list.

2. Extract the tar file using **tar -xvf** command

1.3.1.2 Installation

1. You can choose installation location using **--prefix=path parameter**.

2. You can find **./configure** file in the extracted folder above. For installation follow the below procedure.

3. Inside the extarcted folder, run the following commands one after other

```
./configure --prefix=/scratch/nginx --with-http_ssl_module

make
```

4. You can find the **nginx** installed at **/scratch/nginx**.

1.3.1.3 Start and Stop Procedure

1. Inside the installed directory, we can find sbin folder. Run **./nginx** file inside **sbin** folder to start **nginx**.

2. Write **./nginx -s** stop command in order to stop **nginx**.

1.3.1.4 Configuration:

The configuration is done in the **nginx.conf** file inside the **conf** directory.

1. For Load Balancing, create and upstream and place the URLs to be routed.

```
upstream config{
        server 10.184.155.115:7004;
        server 10.184.148.116:7004;
        }


upstream gateway{
        server 10.184.155.115:7005;
```

Here there are two load balancers defined. Default load balancing technique: **Round Robin**

This block must be placed inside the http block.

ORACLE

2. Configure the Server block. It is also placed inside the http block. The server block is used for routing purposes.

The block is as shown below.

```
server {
        listen       8090;
        server_name  _;

        location / {
                proxy_pass http://config;
        }
        location /api {
                rewrite /api/(.*) /$1  break;
                proxy_pass http://gateway;
        }


        error_page   500 502 503 504  /50x.html;
```

Here 8090 acts as the listen port for **nginx**.

The **server_name** property should be given if the requests are from a particular server. Here Server accepts requests from any server.

The **proxy_pass** is used to route the request and the format of the request is defined in the location tag.

The **rewrite** tag is used to manipulate the requests.

NOTE: Make sure the firewall is open for other IPs to pass the request.

ORACLE

1.3.2 Load Balancer Route Configurations

The below configurations gives the route configurations for the load balancer:

NOTE: The underlying syntax and semantics may vary from load balancer to load balancer.

```
        upstream config-service{
              server << IP Or Hostname of the PLATO Configuration
Service 1 >>:<< Port of the PLATO Configuration Service 1 >>;


              server << IP Or Hostname of the PLATO Configuration
Service 2 >>:<< Port of the PLATO Configuration Service 2 >>;


              server << IP Or Hostname of the PLATO Configuration
Service 3 >>:<< Port of the PLATO Configuration Service 3 >>;



              server << IP Or Hostname of the PLATO Configuration
Service N >>:<< Port of the PLATO Configuration Service N >>;


        }


        upstream api-gateway{



              server << IP Or Hostname of the PLATO API Gateway Service
1 >>:<<Port of the PLATO API Gateway Service 1 >>;


              server << IP Or Hostname of the PLATO API Gateway Service
2 >>:<<Port of the PLATO API Gateway Service 2 >>;


              server << IP Or Hostname of the PLATO API Gateway Service
3 >>:<<Port of the PLATO API Gateway Service 3 >>;


                          .

```

ORACLE®

```
        upstream <<Context Root of the PLATO UI APP Shell>> {


            server << IP Or Hostname of the PLATO UI APP Shell 1 >>:<<
Port of the PLATO Configuration Service 1 >>;


            server << IP Or Hostname of the PLATO UI APP Shell 2 >>:<<
Port of the PLATO UI APP Shell 2 >>;


            server << IP Or Hostname of the PLATO UI APP Shell 3 >>:<<
Port of the PLATO UI APP Shell 3 >>;



                            .


                            .


            server << IP Or Hostname of the PLATO UI APP Shell N >>:<<
Port of the PLATO UI APP Shell N >>;



        }



        server {

          listen << PORT OF THE LOAD BALANCER >>;


          server_name  _;


          location /config-service {
            proxy_pass http://config-service;
          }


          location /api-gateway {
            proxy_pass http://api-gateway;
          }
```

ORACLE®

ORACLE®

ANNEXURE - 4