Oracle® Banking Microservices Architecture

ANNEXURE – 4

Release 14.6.1.0.0
F61116-01
August 2022

Oracle Banking Microservices Architecture ANNEXURE – 4, Release 14.6.1.0.0

F61116-01

Copyright © 2018, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government’s use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Purpose iv
Audience iv
Acronyms and Abbreviations iv
List of Topics iv
Related Documents v

1 Setting Up Multi Node (High Availability Architecture)

2 NGINX Load Balancer Services

2.1 Installation and Setup 2-1
2.1.1 Download 2-1
2.1.2 Install 2-1
2.1.3 Start and Stop Procedure 2-2
2.1.4 Configuration 2-2
2.2 Load Balancer Route Configurations 2-3

Index

Preface

- Purpose
- Audience
- Acronyms and Abbreviations
- List of Topics
- Related Documents

Purpose

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

Audience

This guide is intended for WebLogic admin or ops-web team who are responsible for installing OFSS Banking Products.

Acronyms and Abbreviations

The list of the acronyms and abbreviations that are used in this guide are as follows:

Table 1 Acronyms and Abbreviations

Abbreviation	Description
SMS	Security Management System
CMC	Common Core
MOC	Manufacturing Operations Center

List of Topics

This guide is organized as follows:

Table 2 List of Topics

Topics	Description
Setting Up Multi Node (High Availability Architecture)	This topic provides the information about multi node setup.

Table 2 (Cont.) List of Topics

Topics	Description
NGINX Load Balancer Services	This topic provides the information about the NGINX load balancer services.

Related Documents

For more information, refer to the following documents:

- Product Installation Guide

1

Setting Up Multi Node (High Availability Architecture)

This topic describes about the multi node setup (High Availability Architecture).

Configuration Server Related Changes

The below changes are to be made to the PROPERTIES table specified by the configuration server.

For the Discovery Server:

PLATO Discovery Service must have an entry for its entire peer PLATO Discovery Services configured by `eureka.client.serviceUrl.defaultZone`. It contains a comma-separated list of all Peer PLATO Discovery Services.

Additionally, to enable the peer awareness mode for PLATO Discovery Service, we should set the `eureka.client.register-with-eureka` to true.

Table 1-1 Application Parameters - Discovery Server

ID	APPLICATION	PROFILE	LABEL	KEY	VALUE
1	plato-discovery-service	jdbc	jdbc	eureka.client.serviceUrl.defaultZon	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 >>

For the Individual Services:

Each service must have an entry for all PLATO Discovery Services configured by eureka.client.serviceUrl.defaultZone. It contains a comma-separated list of all PLATO Discovery Services.

Table 1-2 Application Parameters - Individual 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

Plato UI Configuration Server Related Changes

For each product registered in **PRODUCT_SERVICES_ENV_LEDGER**, the user must change the URL to indicate the PLATO API gateway service.

Table 1-3 Load Balancer - URL

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

setDomainEnv.sh Related Changes

For all the Micro Services
Individual microservices should now access the PLATO Config Service through the Load Balancer URI. That is, the property is configured on 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 through the Load balancer URI. That is, the property is 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 >>
```

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, remove the following key from the properties table and add in the *setuseroverrides.sh* file.

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

Requirement of Load Balancers

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

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

PLATO Configuration Service
All domain services access the PLATO Configuration Service to retrieve their configurations. In a multi-node deployment where multiple PLATO Configuration Services are deployed, the user need a single URI to access multiple-node deployments of PLATO Configuration Services. This Load Balancer helps us to achieve that functionality.

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

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

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

2
NGINX Load Balancer Services

This topic describes about the installation and configuration of the NGINX load balancer services.

- Installation and Setup
  This topic describes about the installation and setup.
- Load Balancer Route Configurations
  This topic describes about load balancer route configurations.

2.1 Installation and Setup

This topic describes about the installation and setup.

- Download
  This topic provides systematic instructions to download the tar file.
- Install
  This topic provides systematic instructions to install the tar file.
- Start and Stop Procedure
  This topic provides systematic instructions to start and stop procedure of files.
- Configuration
  This topic provides systematic instructions to configure the server.

2.1.1 Download

This topic provides systematic instructions to download the tar file.

1. Download the tar file from nginx.org/download/ Index list.
2. Extract the tar file using tar -xvf command.

2.1.2 Install

This topic provides systematic instructions to install the tar file.

1. Choose the installation location using **--prefix=path parameter**.
2. Find the `./configure` file in the extracted folder above.
3. Inside the extracted folder, run the following commands one after other

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

   The **nginx** is installed at `/scratch/nginx`.

2.1.3 Start and Stop Procedure

This topic provides systematic instructions to start and stop procedure of files.

1. Find the **sbin** folder inside the installed directory.
2. Run **./nginx** file inside **sbin** folder to start **nginx**.
3. Write **./nginx -s** stop command to stop **nginx**.

2.1.4 Configuration

This topic provides systematic instructions to configure the server.

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

1. Create, upstream, and place the URLs to be routed for Load Balancing.

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

upstream gateway{
     server 10.184.155.115:7005;
```

> **Note:**
>
> There are two load balancers defined. Default load balancing technique is **Round Robin**. This block must be placed inside the http block.

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;
```

   - The `8090` acts as the listen port for **nginx**.

- The `server_name` property must be given if the requests are from a particular server. The Server accepts requests from any server.
- The `proxy_pass` is used to route the request and the request format 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.

2.2 Load Balancer Route Configurations

This topic describes about load balancer route configurations.

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

> **Note:**
>
> The underlying syntax and semantics may vary from the 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 >>;
   }
   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;
      }
```

Index

C

Configuration, *2-2*

D

Download, *2-1*

I

Install, *2-1*

L

Load Balancer Route Configurations, *2-3*

N

NGINX Load Balancer Services, *2-1*

P

Plato UI Configuration Server Related Changes, *1-2*

R

Requirement of Load Balancers, *1-3*

S

setDomainEnv.sh Related Changes, *1-2*
Setting Up Multi Node (High Availability Architecture), *1-1*
Start and Stop Procedure, *2-2*