

ANNEXURE-4
Release 14.7.0.0.0
Part No. F75096-01
November 2022

ORACLE[®]
FINANCIAL SERVICES

ANNEXURE-4
Oracle Financial Services Software Limited
Oracle Park
Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India
Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax: +91 22 6718 3001
www.oracle.com/financialservices/

Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 failsafe, 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.

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.

This software or hardware and documentation may provide access to or information on 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. 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.

Table of Contents

- 1. ANNEXURE..... 1-1
 - 1.1 INTRODUCTION 1-1
 - 1.2 HOW TO DO MULTI NODE SETUP(HIGH AVAILABILITY ARCHITECTURE) 1-1
 - 1.2.1 Configuration Server Related Changes: 1-1
 - 1.2.2 Plato UI Configuration Server Related Changes: 1-2
 - 1.2.3 setDomainEnv.sh related changes: 1-2
 - 1.2.4 Requirement of Load Balancers: 1-2
 - 1.3 PLATO LOAD BALANCER SERVICES 1-4
 - 1.3.1 Introduction 1-4
 - 1.3.2 Configuration and Deployment..... 1-4
 - 1.4 NGINX LOAD BALANCER SERVICES 1-5
 - 1.4.1 Installation and Setup 1-5
 - 1.4.2 Load Balancer Route Configurations: 1-6

1. ANNEXURE

1.1 Introduction

This documents is supporting document, while installing PLATO applications you may find reference.

1.2 How to do Multi Node setup(High Availability Architecture)

1.2.1 Configuration Server Related Changes:

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

A) For the Discovery Server:

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

Also 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.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
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 >>

B) For the Individual Services:

Each service should have an entry of all the PLATO Discovery Service's 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:

A) *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.

Here, plato.services.config.uri should point to the URI of the load balancer. The format of the same would be,

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

B) *For the UI APPSHELL,*

UI APPShell should now access the Gateway Service via the Load balancer URI i.e. configured in the server runtime. e.g -Dapigateway.url.

Here apigateway.url should point to the host and port of the load balancer.

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

1.2.4 Requirement of Load Balancers:

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

PLATO API Gateway Service: PLATO API Gateway Service acts as a single point of entry for both 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.

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.

PLATO UI APP SHELL: The PLATO UI App Shell acts as the single user interface for the users. In a multi node deployment where multiple instances of PLATO UI APP SHELL are deployed, we would need a single URI for accessing the multi node deployments of the PLATO UI APP SHELL. This Load Balancer would help us to achieve that functionality.

1.3 PLATO LOAD BALANCER SERVICES

1.3.1 Introduction

Plato Load Balancer Services can be used as an optional Load Balancer for a service when required. It is available as a WAR file and the configuration can be done directly inside the WAR file.

1.3.2 Configuration and Deployment

Open the war file using extraction tools and open the web.xml file.

The web.xml file is as shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:wls="http://xmlns.oracle.com/weblogic/weblogic-web-app"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.oracle.com/weblogic/weblogic-web-app"
  app">
  <servlet>
    <servlet-name>HttpClusterServlet</servlet-name>
    <servlet-class>
      weblogic.servlet.proxy.HttpClusterServlet
    </servlet-class>
    <init-param>
      <param-name>WebLogicCluster</param-name>
      <param-value>HOST_1:PORT_1|HOST_2:PORT_2|HOST_3:PORT_3</param-
value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>HttpClusterServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>HttpClusterServlet</servlet-name>
    <url-pattern>*. *</url-pattern>
  </servlet-mapping>
</web-app>
```

Edit the *HOST:PORT* in the web.xml file. The host and port values should be added according to the requirement.

After editing the web.xml file the WAR file can be deployed directly into the managed server. The *HOST:PORT* of the plato-load-balancer-services would act as the load balancer URL for the mentioned *HOST:PORT* values in the web.xml file.

1.4 NGINX LOAD BALANCER SERVICES

1.4.1 Installation and Setup

Download:

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

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 extracted folder, run the following commands one after other

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

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

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.

Configuration:

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

1. For Load Balancing, create an 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;  
    server 10.184.148.116:7005;  
}
```

Here there are 2 load balancers defined. Default load balancing technique: *Round Robin*

This block must be placed inside the http block.

2. Server block also needs to be configured. This is also placed inside the http block. The server block is used for routing purposes.

The block looks like 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;
    location = /50x.html {
        root    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.

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

rewrite tag is use dto manipulate the requests.

Note : Make sure the firewall is open for other ips to pass the request.

1.4.2 Load Balancer Route Configurations:

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

Please note that 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 >>;

        .
        .

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

    }

    .
    .
```

```
    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;
    }

    location /<<Context Root of the PLATO UI APP Shell>> {

        proxy_pass http://<<Context Root of the PLATO UI APP Shell>>;

    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }

}
```