

Oracle Banking Trade Finance Process Management
Installation Guide
Oracle Banking Installer
Release 14.3.0.1.0
[Dec] [2019]



Table of Contents

- 1. ABOUT THIS MANUAL..... 1-1**
 - 1.1 INTRODUCTION 1-1
 - 1.2 SCOPE..... 1-1
- 2. INSTALLER PRODUCT 2-2**
 - 2.1 PREREQUISITES 2-2
 - 2.2 DATABASE INSTALLATION 2-12
 - 2.3 WEBLOGIC INSTALLATION 2-13



Installation Guide
[Dec] [2019]
Version 14.3.0.1.0
Oracle Banking Installer
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 © [2007], [2019], 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.



1. About this Manual

1.1 Introduction

1.1.1 Purpose

This document provides the procedure to use the installer product.

1.1.2 Audience

This guide is primarily intended for Developers for Oracle Banking products. Some information may be relevant to IT decision makers and users of the application are also included. Readers are assumed to possess basic operating system, network, and system administration skills with awareness of vendor/third-party software's and knowledge of Oracle Banking products.

1.2 Scope

1.2.1 Read Sections Completely

Each section should be read and understood completely. Instructions should never be blindly applied. Relevant discussion may occur immediately after instructions for an action, so be sure to read whole sections before beginning implementation.

1.2.2 Understand the Purpose of this Guidance

The purpose of the guidance is to provide procedure to use Oracle Banking Installer Product.

1.2.3 Limitations

This guide is limited in its scope to provide procedure to use Oracle Banking Installer Product.

2. Installer Product

The following sections require to be completed in a sequence to finish the installation.

2.1 Prerequisites

Below listed are the prerequisites which are required to use the Oracle Banking Installer:

1. Node machines must have Oracle Linux 7 installed with Java version 1.8.0 patch 181.
2. Install the Chef Client on the node machines to use the Oracle Banking Installer. Refer [Installation of Chef Client](#) section for detailed instructions to install chef client.
3. Make sure that the PDB with applicable DB schemas for the product are available.
4. Install the Oracle SQL Client must be installed on the node machine to use the Oracle Banking Installer for the DB SQL execution.
5. Make sure that the required number of WebLogic (12.2.1.3) managed servers and clusters for the product are up & running on SSL with Custom Trust Key Store (jks).
6. Create the data bags required for the product installation. Refer [Data bag Creation](#) section for detailed instructions to create the data bags.
7. Set bash shell on node machines to run the Oracle Banking Installer.
8. Configure the proxy on each node machine. Set up the proxy with the following commands.
(Proxy setup is required only to update the yum package)

```
$ export http_proxy=http://USERNAME:PASSWORD@proxy-server.mycorp.com:port/
```

```
$ export https_proxy=https://USERNAME:PASSWORD@proxyserver.mycorp.com:port/
```

9. Update yum using the following command.

```
$ yum update yum
```

10. Make sure that the system date & time is valid and latest.
11. FQDN should be set properly. You can check the current FQDN using this command.

```
$ hostname -f
```

2.1.1 Installation of Chef Client

Do the following steps to install the chef client:

1. Login to node machine as root user.
2. Copy the product OSDC bundle to the /scratch folder of the Oracle Linux Server i.e. node machine.
3. Extract the product OSDC bundle and navigate to /scratch/<extracted OSDC bundle>/oracle_banking_installer_pack/softwares/chef-client/14.8.12

```
$ cd /scratch/<<extracted OSDC bundle>>/oracle_banking_installer_pack/softwares/chef-client/14.8.12
```

4. Execute the below command to install the chef-client

```
$ rpm -ivh chef-14.8.12+20190122094824-1.el7.x86_64.rpm
```

5. Verify chef 14.8.12 is installed in the machine using this command.

```
$ chef-client --version
```

2.1.2 Data bag Creation

The data bag contains the encrypted passwords for the WebLogic, DB and SSL. These passwords will be used by the Oracle Banking Installer during the product installation.

Chef requires a secret key to encrypt and decrypt the data bag items. Install the openssl in the node machine and use the following command to generate a secret key:

```
$ openssl rand -base64 512 > /scratch/<extracted OSDC bundle>/oracle_banking_installer_pack/chef-repo/secrets/secret_key
```

Note: Open ssl is one of the option to generate the secret key.

2.1.2.1 WebLogic

1. Navigate to `/scratch/<extracted OSDC bundle>/oracle_banking_installer_pack/chef-repo/data_bags/fsgbu_weblogic_deploy/` folder.
2. Edit the json file "datasource_credentials.json". This json file will contain the data source passwords.

```
$ cd fsgbu_weblogic_deploy
$ vi datasource_credentials.json
```

3. Update plain data source passwords against the corresponding data source name in the json file.

```
{
  "id": "datasource_credentials",
  "<datasource name>": "<password>",
  "<datasource name>": "<password>",
  "<datasource name>": "<password>",
  "<datasource name>": "<password>"
}
```

Note :

For OBTFPM the below mentioned Data sources are mandatory to be available in the commoncore_mw_deploy.rb
domain_services_mw_deploy.rb

S.No	Section	JNDI description
1	plato	Map the schema which created for Plato service - schema
2	plato_security	Map the schema which created for Plato security service - schema

3	plato_ui_config	Map the schema which created for Plato UI service - schema
4	sms	Map the schema which created for SMS service - schema
5	obtfpmadapter	Map the schema which created for Trade Adaptor service - schema
6	obtfpmcommonds	Map the schema which created for Trade Common service - schema
7	obtfpmdoccollectds	Map the schema which created for Trade Documentary collection service - schema
8	obtfpmdrawingsds	Map the schema which created for Trade drawings service - schema
9	obtfpmsmgmt	Map the schema which created for Data segment management service - schema
10	obtfpmextsysrepdata	Map the schema which created for Trade External repository service - schema
11	obtfpmgw	Map the schema which created for Trade Gateway service - schema
12	obtfpmguaranteeds	Map the schema which created for Trade Guarantee service - schema
13	obtfpmlettercreditds	Map the schema which created for Trade Letter Of Credit service - schema
14	obtfpmmomaintenance	Map the schema which created for Maintenance service - schema
15	obtfpmorchestrator	Map the schema which created for Trade Orchestration service - schema
16	obtfpmstagemanagemnt	Map the schema which created for Stage transaction management service - schema
17	obtfpmtemplate	Map the schema which created for Trade Template service - schema
18	obtfpmutility	Map the schema which created for Trade Utility service - schema

4. For reference, see the below example datasource_credentials.json with sample content.

```
{
  "id": "datasource_credentials",
  "plato": "welcome1",
  "plato_security": "welcome1",
  "plato_ui_config": "Password1",
  "plato_task_server": "Password1"
```

```
}
```

5. Execute the below knife command from chef-repo directory to encrypt the passwords.

```
knife data bag from file fsgbu_weblogic_deploy datasource_credentials.json --  
secret-file /scratch/<extracted OSDC  
bundle>/oracle_banking_installer_pack/chef-repo/secrets/secret_key --local-  
mode
```

6. Encrypted data bag file look like below (for reference).

```
{  
  
  "id": "datasource_credentials",  
  
  "plato": {  
  
    "encrypted_data":  
"m0suBokKsy8Bu5X33YFd39JAbqz5CTrB4gVd9FwNuEKvYsHzbkHQyUpTHEkG\n5aI+ka/z2Pcddt1  
zFgp4evnp4ERpFWqk7FLBf11TfYpcs3aRnC5MkSMmt208\nihr/C9ZcUdI/F6rnA4vJEpnmpyHkM1M  
fGb5FaZDVgcNvz3gDJLrMt9WEGtqf\nnr0+UU7AjMs/bo011Jj9Lm09T71UPhOPmMLpxSrvKqAEz3ia  
VWwJayC1IfxXy\nwuXhV+hR0HuP\n",  
  
    "iv": "Y0Iafm7EWhmjIjX1\n",  
  
    "auth_tag": "+Y68ZesLqsiYkoI2KyIREQ==\n",  
  
    "version": 3,  
  
    "cipher": "aes-256-gcm"  
  
  },  
  
  "plato_security": {  
  
    "encrypted_data":  
"SSCivRfo+B8LIgtDd48fTiOXGam2W6J2RXZrYRLPIJDoM0kjb8UGOLj116/\nDdGpZcu5c6UnbF1  
AwRqX6OTdLS4GBnk3WhqBAaFXwSpad8eh5MNKofPZJ300\nnmz6fJVz7xdxU78LKcgxOMI1XALSHeSx  
/1atXHxLgVqAM8eXjpG6vxwYlzmch\nnsb5jnZ0WsoRjCumaJ0j/ohRUFwv6Eyrv5ESjj35qOfnB9I1  
qDZzz/HN8A/A0\nCXirTNM=\n",  
  
    "iv": "4dkvfbyRbjvK0Tk8\n",  
  
    "auth_tag": "r45/oPUZ9Yd0FTp+mwV44Q==\n",  
  
    "version": 3,  
  
    "cipher": "aes-256-gcm"  
  
  },  
  
  "plato_ui_config": {
```



```

    "encrypted_data":
    "bukZXxOruMZw4JRoyAqHazUa7XEdRD3gr7TUnwppnGwki30/cjdBdE5unL6E\nXgcM8Jp08zfLfWS
    EJKn62N8spH7GVlc5A0t+TA5vLeHj2B5PKDtxGYPcLJXA\nx8T3SpYQrSUxazwKcZzC7rsSFKpci6N
    km7N4pM08qPAONwRUconbGevs9B1k\nRg7WFhDi4o+pVGJf3+lqEt12xmillyP888cpf2ttQktJOH1
    4aXZWGyQxKo8u\n8h0iM0N0Ksaq\n",

    "iv": "m+2ia1bYTfAhbbq8\n",

    "auth_tag": "J6txcDNmPRcYkIXus3XWvg==\n",

    "version": 3,

    "cipher": "aes-256-gcm"
  },

  "plato_task_server": {

    "encrypted_data": "s0i20txVXx2N7c7WgfMysvn9k04B2SzzaH02+7KJZTHR\n",

    "iv": "vLIF30pVZBLyucPr\n",

    "auth_tag": "jKEXypt170YSu3y9Wcun0Q==\n",

    "version": 3,

    "cipher": "aes-256-gcm"
  }
}

```

7. Edit the json file “ssl_credentials.json”. This json file will contain the Custom TrustKeyStore Passphrase.

```

$ cd fsgbu_weblogic_deploy
$ vi ssl_credentials.json

```

8. Update plain Custom TrustKeyStore Passphrase against “ssl_password” key in the json file.

```

{

  "id": "ssl_credentials_",

  "ssl_password": "<Custom TrustKeyStore Passphrase>"

}

```

9. For reference, see the below example ssl_credentials.json with sample content.

```

{

  "id": "ssl_credentials",

```

```
"ssl_password": "welcome1"
}
```

10. Execute the below knife command from chef-repo directory.

```
knife data bag from file fsgbu_weblogic_deploy ssl_credentials.json --secret-  
file /scratch/<extracted OSDC bundle>/oracle_banking_installer_pack/chef-  
repo/secrets/secret_key --local-mode
```

11. Encrypted data bag file look like below (for reference).

```
{
  "id": "ssl_credentials",
  "ssl_password": {
    "encrypted_data": "ZsmK4tgXb4L1NN1w0hAtDqVrFmVJaxa/1Wxm7Q==\n",
    "iv": "tTrnDbFSDmpeeap+\n",
    "auth_tag": "zdNDBY1tcSQe2lozkeC16g==\n",
    "version": 3,
    "cipher": "aes-256-gcm"
  }
}
```

12. Edit the json file “weblogic_credentials.json”. This json file will contain the WebLogic admin username and password.

```
$ cd fsgbu_weblogic_deploy
$ vi weblogic_credentials.json
```

13. Update plain text WebLogic admin username and password against “wl_admin_username” and “wl_admin_password” keys respectively in the json file.

```
{
  "id": " weblogic_credentials",
  "wl_admin_password": "<weblogic admin user password>",
  "wl_admin_username": "<weblogic admin user name>"
}
```

14. For reference, see the below example weblogic_credentials.json with sample content.

```
{
  "id": " weblogic_credentials",
```

```

"wl_admin_password": "welcome1",

"wl_admin_username": "weblogic"

}

```

15. Execute the below knife command from chef-repo directory.

```

knife data bag from file fsgbu_weblogic_deploy weblogic_credentials.json --
secret-file /scratch/<extracted OSDC
bundle>/oracle_banking_installer_pack/chef-repo/secrets/secret_key --local-
mode

```

16. Encrypted data bag file look like below (for reference)

```

{
  "id": " weblogic_credentials",
  "wl_admin_password": {
    "encrypted_data": "vauqQm/Imoig1u7XW8ciWnRDkVk7LM+p8Hs98g==\n",
    "iv": "BoN0RFK7KBnVqcSM\n",
    "auth_tag": "ozvF+Weo7sfVIE0a/tgfMA==\n",
    "version": 3,
    "cipher": "aes-256-gcm"
  },
  "wl_admin_username": {
    "encrypted_data": "n1VnxPa1GcUp9jEheZKX4CgkDw6hP7hXwghk\n",
    "iv": "ibZRT+G1JTqKsTS4\n",
    "auth_tag": "FXurd63q3yfFwU1xU0JSXA==\n",
    "version": 3,
    "cipher": "aes-256-gcm"
  }
}

```

17. Make sure all json files are on 755 permission.

2.1.2.2 Database

1. Navigate to /scratch/<extracted OSDC bundle>/oracle_banking_installer_pack/chef-repo/data_bags/fsgbu_db_deploy folder.
2. Edit the json file "schema_credentials.json". This json file contains the db login passwords.

```
$ cd fsgbu_db_deploy
$ vi schema_credentials.json
```

Note :

For the OBTFPM installation the below schema details are mandatory to maintain in configuration mapping

S.No	Schema Mapping ID	Schema Name	Description
1	COMMON_CORE_SCHEMA_USERNAME	CMNCORE	Map the schema which created for Common core
2	SMS_SCHEMA_USERNAME	SMS	Map the schema which created for SMS
3	PLATO_SCHEMA_USERNAME	PLATO	Map the schema which created for Plato
4	ADAPTER_SERVICES_SCHEMA_USERNAME	TFADPR	Map the schema which created for Trade Adaptor service
5	PLATO_SECURITY_SCHEMA_USERNAME	PLATOSEC	Map the schema which created for Plato security
6	COMMON_DATA_SEGMENTS_SERVICES_SCHEMA_USERNAME	TFCMNDS	Map the schema which created for Trade Common services
7	PLATO_UI_CONFIG_SCHEMA_USERNAME	PLATOUI	Map the schema which created for Plato UI service
8	DOCUMENTARY_COLLECTIONS_SERVICES_SCHEMA_USERNAME	TFDOCCOL	Map the schema which created for Trade Documentary collection service
9	DRAWINGS_SERVICES_SCHEMA_USERNAME	TFDRAWINGS	Map the schema which created for Trade drawings service
10	DATA_SEGMENTS_SERVICES_SCHEMA_USERNAME	TFDSMGMT	Map the schema which created for Data segment management service
11	EXTERNAL_SYS_REPLICATED_SERVICES_SCHEMA_USERNAME	TFEXTREP	Map the schema which created for Trade External repository service
12	GUARANTEE_SERVICES_SCHEMA_USERNAME	TFGTEES	Map the schema which created for Trade Guarantee service
13	LETTER_OF_CREDIT_SERVICES_SCHEMA_USERNAME	TFLC	Map the schema which created for Trade Letter Of Credit service
14	GATEWAY_SERVICES_SCHEMA_USERNAME	TFGATEWAY	Map the schema which created for Trade Gateway service
15	STAGE_MANAGEMENT_SCHEMA_USERNAME	TFSTGMT	Map the schema which created for Stage transaction management service
16	MAINTENANCE_SERVICES_SCHEMA_USERNAME	TFMAIN	Map the schema which created for Maintenance service
17	TEMPLATE_SERVICES_SCHEMA_USERNAME	TFTEMP	Map the schema which created for Trade Template service
18	ORCHESTRATOR_SERVICES_SCHEMA_USERNAME	TFORCH	Map the schema which created for Trade Orchestration service
19	UTILITY_SERVICES_SCHEMA_USERNAME	TFUTILITY	Map the schema which created for Trade Utility service

- Update the database passwords against their respective place holders in the json file. Schema Passwords should be updated against their respective schema names.

```
{
  "id": "schema_credentials",
```

```

"<schema name>_schema_password": "<schema password>",
"<schema name>_schema_password": "<schema password>"
}

```

4. For reference, see the below example schema_credentials.json with sample content.

```

{
  "id": " schema_credentials",
  "PLATO_SCHEMA_PASSWORD": "fsgbu",
  "PLATO_SECURITY_SCHEMA_PASSWORD": "fsgbu"
}

```

5. Execute the below knife command from chef-repo directory.

```

knife data bag from file fsgbu_db_deploy schema_credentials.json --secret-file
/scratch/<extracted OSDC bundle>/oracle_banking_installer_pack/chef-repo/secrets/secret_key -
-local-mode

```

6. Encrypted data bag file look like below (for reference).

```

{
  "id": "schema_credentials",
  "PLATO_SCHEMA_PASSWORD": {
    "encrypted_data": "66xvqPDzuEBPNjXj0HFNfGCPK0nIikCovHtXvLQFRxtAe2Q=\n",
    "iv": "ib3jAYpHcU54Tz1s\n",
    "auth_tag": "mveskCvVZYjqKVEN2CKNeQ==\n",
    "version": 3,
    "cipher": "aes-256-gcm"
  },
  "PLATO_SECURITY_SCHEMA_PASSWORD": {
    "encrypted_data": "L2Nk1cTNqYdFKG10CYhwceunnuQoRuxDJJz1MQp1a9D0\n",
    "iv": "WfQ9hCyjJKg1zwko\n",
    "auth_tag": "/ZpEvNuEo+e3d0B/rLtZfg==\n",
    "version": 3,
    "cipher": "aes-256-gcm"
  }
}

```

```
}  
}
```

7. Make sure all json files are on 755 permission.

2.2 Database Installation

Note: Make sure that all the DB schemas which are required for the product are available before installation.

1. Login to Linux server as root user
2. Go to /scratch/<extracted OSDC bundle>/oracle_banking_installer_pack/chef-repo folder.
3. Update Parameter file "**db_deploy_properties.rb**" with relevant values as described in below table.

Attribute	Description
ORACLE_SERVER_HOSTNAME	Oracle Database Server Hostname
ORACLE_CLIENT_INSTALL_USER	User that owns the Oracle home directory.
ORACLE_CLIENT_HOME	Path to Oracle database Client home
PDB_SID	PDB SID
PDB_PORT	PDB Port
PRODUCT_CUSTOM_SCRIPTS_HOME	Path to the product directory inside the OSDC folder. Eg:- /scratch/OSDC/<product directory>
PLATO_SCHEMA_USERNAME	Plato schema username
PLATO_SECURITY_SCHEMA_USERNAME	Plato security schema username

4. Execute the shell script **db_deploy_installer.sh** under /oracle_banking_installer_pack/chef-repo
5. Tail the log file to see the progress

```
$ tail -f nohup.out
```

6. Once the db_installer script run complete, all SQL files from the following location must be compiled in the respective DB schema:

```
/scratch/<extracted OSDC bundle>/OBTfPM_SERVICES/obtfpm-template-services/DB/DOMAIN/SQL
```

```
/scratch/<extracted OSDC bundle>/OBTfPM_SERVICES/cmc-additional-attributes-services/DB/DOMAIN/SQL
```

7. Check the properties table in PLATO schema for the port and URL related properties are referring to the correct server/port.

2.3 WebLogic Installation

Note: Make sure that all the DB schemas which are required for the product are available and make sure that the required number of WebLogic managed servers and clusters for the product are up & running on configured SSL before installation.

1. Login to Linux server as root user
2. Go to /scratch/<extracted OSDC bundle>/oracle_banking_installer_pack/chef-repo folder.
3. Make sure that the weblogic install user has write permission on the extracted OSDC bundle.
4. Update Parameter file “*pre_installation_properties.rb*” with relevant values as described in below table

Attribute	Description
PRODUCT_BUNDLE_HOME	Path to the product directory inside the OSDC folder. E.g.:- /scratch/OSDC/<product directory>.
APP_STARTERS_PATH_URL	Path to the destination folder inside which App starters files should be copied.
TASKS_PATH_URL	Path to the destination folder inside which Task files should be copied.

5. Execute the shell script ***pre_installation_package_mw_deploy.sh*** under /oracle_banking_installer_pack/chef-repo
6. Tail the log file to see the progress

```
$ tail -f nohup.out
```

7. Update Parameter file “*mw_deploy_properties.rb*” with relevant values as described in below table.

Attribute	Description
JAVA_HOME	Path to the directory where java is installed.
WLS_HOME	Path to the WebLogic home directory.
WLS_INSTALL_USER	User who owns the WebLogic home directory

WLS_INSTALL_USER_GROUP	Group that owns the WebLogic home directory.
WLS_DOMAIN_NAME	Name of the WebLogic domain
IS_TARGET_CLUSTER	Whether the targets to which the deployment is to be done are clusters (true/false)
WLS_SSL_CUSTOM_TRUST_FILE	Location of the SSL custom trust file
WLS_SSL_CUSTOM_TRUST_KEYSTORE_TYPE	SSL Trust KeyStore Type(JCEKS or JKS)
WEBLOGIC_ADMIN_HOST	Weblogic Admin Host
WEBLOGIC_ADMIN_LISTEN_PORT	WebLogic Admin listen port
WEBLOGIC_ADMIN_SSL_PORT	WebLogic Admin SSL port
ORACLE_PDB_SID	Oracle PDB SID for data source configuration
ORACLE_PDB_HOSTNAME	Oracle PDB hostname for data source configuration
ORACLE_PDB_PORT	Oracle PDB port for data source configuration
PRODUCT_BUNDLE_HOME	Path to the product directory inside the OSDC folder. E.g.:- /scratch/OSDC/<product directory>
PLATO_DS_USERNAME	Plato datasource username
PLATO_SECURITY_DS_USERNAME	Plato security datasource username
PLATO_TASK_SERVER_DS_USERNAME	Plato task server datasource username
PLATO_UI_CONFIG_DS_USERNAME	Plato UI Config datasource username
PLATO_DS_TARGET	Plato datasource target
PLATO_SECURITY_DS_TARGET	Plato security datasource target

PLATO_UI_CONFIG_DS_TARGET	Plato UI config datasource target
PLATO_TASK_SERVER_DS_TARGET	Plato task server datasource target
SMS_DS_TARGET	SMS datasource target
COMMON_CORE_DS_TARGET	Commoncore datasource target
DISCOVERY_SVCS_TARGET	Discovery services target
CONFIG_SVCS_TARGET	Config services target
API_GATEWAY_TARGET	API gateway target
UI_CONFIG_TARGET	UI config target
TASK_SERVER_TARGET	Task server target
SMS_CORE_SVCS_TARGET	SMS core services target
COMMON_CORE_TARGET	Commoncore services target

8. Execute the shell scripts under /oracle_banking_installer_pack/chef-repo in the following order

- 1) plato_mw_deploy_installer.sh
- 2) commoncore_mw_deploy_installer.sh
- 3) domain_services_mw_deploy_installer.sh
- 4) *midoffice_commoncore_mw_deploy_installer.sh*

9. Tail the log file to see the progress

```
$ tail -f nohup.out
```