



ORACLE

Oracle Application Express (APEX) on Oracle Cloud Infrastructure Database

Terraform Deployment

March 2021, Version 2.0
Copyright © 2021, Oracle and/or its affiliates
Public

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Revision History

The following revisions have been made to this document:

DATE	REVISION
March 19, 2021	<ul style="list-style-type: none">Updated the link to download the Terraform template and environment setup shell script.Updated the template.
March 15, 2019	Initial publication

Table of Contents

Introduction	4
Overview	4
Installation	5
Step 1: Download the Terraform Script and Configure the Local System for Shell and Terraform	5
Step 2: Run Setup.sh to Set Environment Variables	5
Step 3: Perform Prerequisite Tasks	11
Step 4: Run the Terraform Script	13
Step 5: Access Oracle REST Data Services	14
Maintenance	14
Conclusion	15
References	15

Introduction

Oracle Application Express (APEX) is a low-code development platform that enables you to build stunning, scalable, secure apps with world-class features that can be deployed anywhere. Using only a web browser and limited programming experience, you can use APEX to rapidly develop and deploy professional applications that are both fast and secure for any device, from desktop to mobile.

Until now, customers have had to perform standard installation steps to install Oracle APEX on Oracle Cloud Infrastructure Database instances. To simplify this installation, Oracle has released a Terraform template that fully automates Oracle APEX installation in a customer tenancy. This document provides step-by-step guidance for deploying Oracle APEX on Oracle Cloud Infrastructure Database by using the Terraform template. Although the manual installation process is still supported, the Terraform template simplifies the process while implementing best practices for Oracle APEX deployment.

This document assumes that you have a basic understanding of various components of Oracle Cloud Infrastructure and Oracle APEX, as well as Terraform. For information, see the following resources:

- [Overview of Oracle Cloud Infrastructure](#)
- [Oracle Cloud Infrastructure Networking](#)
- [Oracle APEX](#)
- [Getting Started with the Terraform Provider](#)

Overview

Using the steps provided in this document, you deploy Oracle APEX on the target database and Oracle REST Data Services as a web listener on a compute VM in your tenancy. At a high level, the Terraform template performs the following actions:

- Creates a compute instance on a public subnet in a VCN
- Downloads the relevant binaries to the compute VM: `ords.war`, JDK, Tomcat (if selected)
- Installs the downloaded binaries
- Creates and configures the Oracle APEX database schema
- Performs SSL configuration on the web container

Following are some advantages of deploying Oracle REST Data Services (sometimes referred to as ORDS) on a separate compute instance:

- Allows access to the Oracle APEX instance and Oracle REST Data Services even if the target databases are in private networks, removing the requirement to have a public IP on the database instance.
- Makes it easy to patch and upgrade Oracle REST Data Services independent of the target databases. To upgrade or patch Oracle REST Data Services, you can simply stop (or terminate) the Oracle REST Data Services VM and run the Terraform template again with the latest version of Oracle REST Data Services.
- Use a single installation of Oracle REST Data Services with multiple database instances.

In line with Oracle Cloud Infrastructure security standards, the Terraform script supports SSL/HTTPS configuration to access Oracle REST Data Services deployment. You can use IP-based or FQDN-based browser access. IP-based access uses SSL via self-signed certificates, and FQDN-based access uses Let's Encrypt certificates for SSL encryption. We recommend using FQDN-based access.

Note: Oracle APEX is a no-cost feature of Oracle Database and is fully supported for Oracle Cloud Infrastructure Database deployment. Support for the Terraform template is on a best-effort basis.

Installation

This section provides instructions for using the Terraform script to deploy Oracle APEX on the target database and Oracle REST Data Services as the web listener on a compute VM in your tenancy.

Step 1: Download the Terraform Script and Configure the Local System for Shell and Terraform

1. Download the [Terraform template and environment setup shell script](#).
2. Configure your laptop or desktop for Terraform and shell.

[Learn more about Terraform and the Oracle Cloud Infrastructure Terraform provider.](#)

Step 2: Run Setup.sh to Set Environment Variables

Run the Setup.sh shell script in Bash shell (or using a relevant command prompt). Based on the options that you select, you need values for some or all the following variables before running the script.

VARIABLE	DESCRIPTION
TF_VAR_region	Oracle Cloud Infrastructure region key
TF_VAR_AD	Availability domain (AD) number 1 2 3 (logical availability domain)
TF_VAR_InstanceOSVersion	Oracle Linux 7.x version available on Oracle Cloud Infrastructure
TF_VAR_tenancy_ocid	Tenancy's Oracle Cloud identifier (OCID)
TF_VAR_user_ocid	User's OCID
TF_VAR_fingerprint	Fingerprint for API access
TF_VAR_private_key_path	Path to the API private key
TF_VAR_ssh_public_key	SSH public key
TF_VAR_ssh_private_key	SSH private key
TF_VAR_api_private_key	API private key
TF_VAR_compartment_ocid	Compartment's OCID
TF_VAR_subnet_ocid	Subnet's OCID
TF_VAR_target_db_ip	Private IP address of the node for single instances, or SCAN name for RAC instances
TF_VAR_target_db_srv_name	Service name of the database (for Oracle Multitenant database, the service name of the PDB) Note: We don't recommend installing Oracle APEX common in CDB\$ROOT.
TF_VAR_web_srv	Flag for the web container to be configured: 0 for Jetty, 1 for Tomcat
TF_VAR_URL_ORDS_file	URL of Object Storage for the ords.war file
TF_VAR_ComputeDisplayName	Display name of the compute instance
TF_VAR_InstanceName	Host name of the compute instance (not including the domain name)

VARIABLE	DESCRIPTION
TF_VAR_InstanceShape	Shape of the compute VM instance Note: Oracle REST Data Services is a lightweight tool that doesn't require many resources to run. However, the required shape depends on the number of concurrent connections. We recommend starting small and then rebuilding with a larger shape if needed.
TF_VAR_com_port	Port to be open to access the web server on the compute instance
TF_VAR_Secure_FQDN_access	Flag for FQDN access: 0 for FQDN access, 1 for IP address access
TF_VAR_URL_APEX_file	URL of the Oracle APEX zip file
TF_VAR_APEX_install_mode	Flag for Oracle APEX installation mode: 0 for full development environment, 1 for runtime environment
TF_VAR_URL_tomcat_file (optional)	URL of Object Storage for the Tomcat 8.5 tar .gz extension file
TF_VAR_ZoneName (optional)	Zone name for the DNS domain

Sample Output of the Setup.sh Script

```

$ ./setup.sh
-----
Setting up environment variables to launch a compute instance for ORDS.
Please enter required information below:
-----

**** Authentication ****
Enter your tenancy's OCID []: ocid1.tenancy.oc1..
Enter your user's OCID []: ocid1.user.oc1..
Enter your fingerprint []: 29:2a:a4:5c:35:5e:3c:12:f6:50:6f:79:5c:89:b8:5c
Enter path to your SSH public key []: $HOME/.ssh/id_rsa.pub
Enter path to your SSH private key []: $HOME/.ssh/id_rsa
Enter path to your API private key []: $HOME/.oci/oci_api_key.pem
Enter your compartment's OCID []: ocid1.compartment.oc1..

**** Target Database ****
Enter the database host's IP address []: 10.0.0.24
Enter the database service name []: jetty_fqdn.sub03020528440.test.oraclevcn.com

**** Compute Instance ****
Enter the region <us-phoenix-1|us-ashburn-1|eu-frankfurt-1|uk-london-1> []: uk-london-1
Enter the AD <1|2|3> []: 3
Enter the available Oracle Linux 7.X version <7.9 |...>[]:
Enter the subnet's OCID []: ocid1.subnet.oc1.uk-london-1.
Enter the Compute Instance's display name []: ORDS-Comp
Enter the Compute Instance's hostname []: ords-comp
Enter the Compute Instance's shape <VM.Standard2.1|VM.Standard2.2|...> []: VM.Standard2.1
Enter the Compute Instance's port []: 8888
Enter 0 to run Jetty (ORDS standalone mode), or 1 to run Tomcat [0]: 0

```

```
Enter 0 to access with FQDN (hostname.yourdomain), or 1 to access with public IP address
[1]: 1
```

```
***** File location on Object Storage *****
```

```
Enter the URL for ORDS.war file []: https://objectstorage.uk-london-1.oraclecloud.com/
```

```
Enter the URL for APEX zip file []: https://objectstorage.uk-london-1.oraclecloud.com/
```

```
***** APEX installation mode *****
```

```
Enter 0 to install APEX with Full development environment mode, or 1 to install APEX with
Runtime environment mode [0]: 0
```

```
INFO: Updating values...
```

```
INFO: env-vars is updated successfully.
```

```
    Please set environment variables. (ex. source env-vars)
```

```
$
```

```
$ cat env-vars
```

```
### Path to keys
```

```
PathToYourSshPublicKey=$HOME/.ssh/id_rsa.pub
```

```
PathToYourSshPrivateKey=$HOME/.ssh/id_rsa
```

```
PathToYourApiPrivateKey=$HOME/.oci/oci_api_key.pem
```

```
### Authentication details
```

```
export TF_VAR_tenancy_ocid=ocid1.tenancy.oc1..
```

```
export TF_VAR_user_ocid=ocid1.user.oc1..
```

```
export TF_VAR_fingerprint=29:2a:a4:5c:35:5e:3c:12:f6:50:6f:79:5c:89:b8:5c
```

```
export TF_VAR_private_key_path=${PathToYourApiPrivateKey}
```

```
### Compartment
```

```
export TF_VAR_compartment_ocid=ocid1.compartment.oc1..
```

```
### Public/private keys used on the instances
```

```
export TF_VAR_ssh_public_key=$(cat ${PathToYourSshPublicKey} 2>/dev/null)
```

```
export TF_VAR_ssh_private_key=$(cat ${PathToYourSshPrivateKey} 2>/dev/null)
```

```
export TF_VAR_api_private_key=$(cat ${PathToYourApiPrivateKey} 2>/dev/null)
```

```
### Log
```

```
# export TF_LOG=TRACE
```

```
# export TF_LOG_PATH='./terraform.log'
```

```
### Target Database
```

```
# export TF_VAR_target_db_admin_pw=
```

```
export TF_VAR_target_db_ip=10.0.0.24
```

```
export TF_VAR_target_db_srv_name=jetty_fqdn.sub03020528440.test.oraclevcn.com
```

```
export TF_VAR_target_db_name=`echo $TF_VAR_target_db_srv_name|awk -F. '{print $1}'`
```

```
### Compute Setting
```

```
# TF_VAR_region : us-phoenix-1 | us-ashburn-1 | eu-frankfurt-1 | uk-london-1
```

```
export TF_VAR_region=uk-london-1
```

```
# TF_VAR_AD : 1 | 2 | 3
```

```
export TF_VAR_AD=3
```

```
export TF_VAR_InstanceOSVersion=7.6
```

```
# TF_VAR_URL_ORDS_file : Object Storage URL for ords.war
```

```
export TF_VAR_URL_ORDS_file=https://objectstorage.uk-london-1.oraclecloud.com/
```

```
# TF_VAR_web_srv : 0 => Jetty, 1 => Tomcat
```

```

export TF_VAR_web_srv=0
# TF_VAR_URL_tomcat_file : Object Storage URL for Tomcat 8.5 tar.gz file, required only when
tomcat is used for Web server
export TF_VAR_URL_tomcat_file=
export TF_VAR_com_port=8888
export TF_VAR_ComputeDisplayName=ORDS-Comp
export TF_VAR_InstanceName=ords-comp
export TF_VAR_InstanceShape=VM.Standard2.1
# TF_VAR_Secure_FQDN_access : 0 => Yes(Enabled FQDN access with CA SSL), 1 => No(Access w/
IP address)
export TF_VAR_Secure_FQDN_access=1

export TF_VAR_ZoneName=
export TF_VAR_URL_APEX_file=https://objectstorage.uk-london-1.oraclecloud.com/
# TF_VAR_APEX_install_mode : 0 => Full Environment mode, 1 => Runtime Environment mode
export TF_VAR_APEX_install_mode=0
export TF_VAR_subnet_ocid=ocid1.subnet.oc1.uk-london-1.

```

Standalone Server (Jetty) or Apache Tomcat for ORDS

The script provides you the option to deploy Oracle REST Data Services in standalone mode (Jetty) or on Apache Tomcat server. For a list of the application servers supported by Oracle REST Data Services, see the [Oracle REST Data Services documentation](#).

```

$ ./setup.sh
-----
Setting up environment variables to launch a compute instance for ORDS.
Please enter required information below:
-----
.
.
.
Enter 0 to run Jetty (ORDS standalone mode), or 1 to run Tomcat []: 1
***** File location on Object Storage *****
Enter the URL for Tomcat 8.5 tar.gz file []: https://objectstorage.uk-london-
1.oraclecloud.com/apache-tomcat-8.5.32.tar.gz

$ cat env-vars
.
.
.
# TF_VAR_web_srv : 0 => Jetty, 1 => Tomcat
export TF_VAR_web_srv=1
# TF_VAR_URL_tomcat_file : Object Storage URL for Tomcat 8.5 tar.gz file, required only when
tomcat is used for Web server
export TF_VAR_URL_tomcat_file=https://objectstorage.uk-london-1.oraclecloud.com

```

IP-Based Browser Access

You can set up either FQDN-based or IP-based browser access. The Terraform script is configured such that IP-based access is configured with self-signed certificates for HTTPS. FQDN-based access uses Let's Encrypt certificates to configure HTTPS for Oracle REST Data Services. You can also bring your own certificates to configure SSL.

```
$ ./setup.sh
-----
Setting up environment variables to launch a compute instance for ORDS.
Please enter required information below:
-----
.
.
.
Enter 0 to access with FQDN (hostname.yourdomain), or 1 to access with public IP address []:
1

$ cat env-vars
.
.
.
# TF_VAR_Secure_FQDN_access : 0 => Yes(Enabled FQDN access with CA SSL), 1 => No(Access w/
IP address)
export TF_VAR_Secure_FQDN_access=1
export TF_VAR_ZoneName=
```

FQDN-Based Browser Access

To enable FQDN-based browser access, you must select the 0 option when running the environment setup shell script.

FQDN-based access uses the free but authorized certification provider Let's Encrypt for SSL. Let's Encrypt is one of many options available in the market, and if you want to use another certificate provider, you can do so.

To automate the process of enabling SSL, the Terraform script uses the [acme.sh certificate](#) management agent listed on the Let's Encrypt [ACME Client Implementations page](#). The process for acquiring the certificate is explained on the [How It Works page](#).

As described on the How It Works page, there are two options for getting a certificate:

- Provisioning a DNS record under `example.com`, or (dns-01)
- Provisioning an HTTP resource under a well-known URI on `https://example.com/`, or (http-01)

The second option requires port 80 or 443 to be opened on the compute instance. This requirement might be a blocker for many customers because of security-related concerns. Therefore, the Terraform script provisions a DNS record under `example.com`, or (dns-01). The script creates DNS resources and takes care of domain and record setting.

A certificate from Let's Encrypt is valid for three months. Using cron and ACME protocol clients, automating the certificate renewal is recommended.

```
$ ./setup.sh
-----
Setting up environment variables to launch a compute instance for ORDS.
Please enter required information below:
-----
.
.
.
Enter 0 to access with FQDN (hostname.yourdomain), or 1 to access with public IP address []:
1

$ cat env-vars
.
.
.
# TF_VAR_Secure_FQDN_access : 0 => Yes(Enabled FQDN access with CA SSL), 1 => No(Access w/
IP address)
export TF_VAR_Secure_FQDN_access=0
export TF_VAR_ZoneName=oci-dbaas.tk
```

Full Development Environment Mode or Runtime Environment Mode

You choose the Oracle APEX installation mode: full development environment or runtime environment.

- A full development environment provides complete access to the App Builder development environment to develop applications.
- A runtime environment is an appropriate choice for production implementations in which you want to run applications that can't be modified.

```
$ ./setup.sh
-----
Setting up environment variables to launch a compute instance for ORDS.
Please enter required information below:
-----
.
.
.
***** APEX installation mode *****
Enter 0 to install APEX with Full development environment mode, or 1 to install APEX with
Runtime environment mode [0]:

$ cat env-vars
.
.
.
# TF_VAR_APEX_install_mode : 0 => Full Environment mode, 1 => Runtime Environment mode
export TF_VAR_APEX_install_mode=0
```

Step 3: Perform Prerequisite Tasks

Before you run the script, perform the following tasks:

- Download the latest versions of [Oracle REST Data Services](#) and [Oracle APEX](#) and place them in an [Object Storage bucket](#).
- Configure the network: a virtual cloud network (VCN), subnets, and security lists.
 - [Create a VCN with a public subnet](#) for the compute VM where Oracle REST Data Services will be deployed.

Note: You can deploy Oracle REST Data Services on a private subnet, but then it can't be accessed over the internet unless you also deploy a load balancing instance. The Terraform script isn't designed for this scenario, however, and won't complete successfully. To install Oracle REST Data Services on a private subnet, you must use a NAT gateway and a service gateway to meet access requirements for various resources. This approach requires advanced Oracle Cloud Infrastructure networking knowledge.

- Create an [internet gateway](#). An internet gateway is needed to access the compute VM via a public IP address to deploy Oracle REST Data Services on it.
- Create [security list rules](#).
- Create an ingress rule on the VM compute subnet for the port (TF_VAR_com_port) on which Oracle REST Data Services is listening.
- Create an ingress rule on the target database subnet for the port on which the database is listening.

Note: Your database instance can be on a private subnet, but the compute VM on which Oracle REST Data Services is deployed must be able to reach the database instance.

- Ensure that sufficient limits are available for the compute VM shape on which Oracle REST Data Services will be deployed.
- [Create a target database](#) if one does not exist. You can use this script to install Oracle APEX on all supported versions of Oracle Cloud Infrastructure Database VM, bare metal, and Exadata databases.

Note: You can't use this Terraform template with Oracle Autonomous Transaction Processing or Oracle Autonomous Data Warehouse because of the SYS access requirement. Oracle APEX support is planned for a future release.

- Ensure that you have SYS access to the target database.

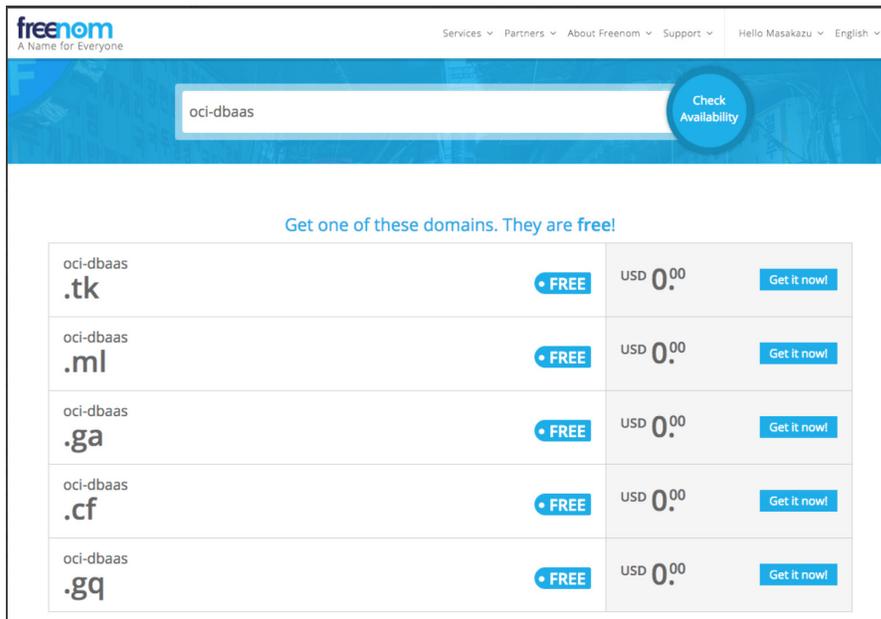
Note: In an Oracle Multitenant environment, the connection string should be to the PDB.

- (Optional) Download the latest version of Tomcat and place it on Object Storage.
- (Optional) Get a valid external domain, which is required to configure FQDN-based access. (See the following section.)

Getting a Valid External Domain

These steps get a free domain from <https://freenom.com/>. However, you can get a domain from any DNS provider that you choose.

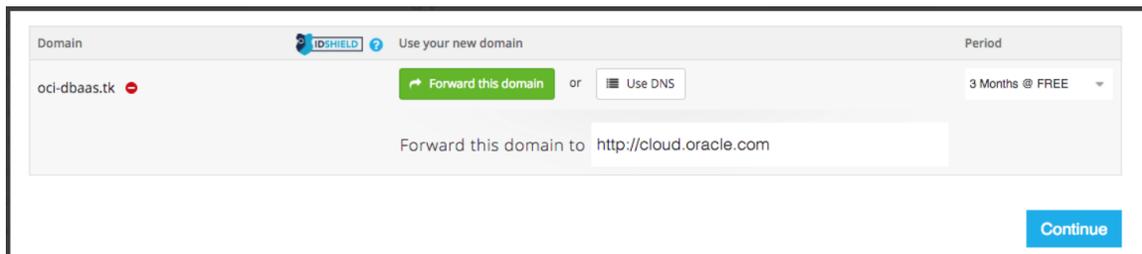
1. Go to <https://freenom.com/> and sign in or sign up.
2. Verify that the domain you want is available by using the search box on the home page or by selecting **Services > Register a New Domain** after signing in.



The screenshot shows the Freenom website interface. At the top, there is a search bar with the text 'oci-dbaas' and a 'Check Availability' button. Below the search bar, a message reads 'Get one of these domains. They are free!'. A table lists five domain options, each with a 'FREE' badge, a price of 'USD 0.00', and a 'Get it now!' button.

Domain	Price	Action
oci-dbaas .tk	USD 0.00	Get it now!
oci-dbaas .ml	USD 0.00	Get it now!
oci-dbaas .ga	USD 0.00	Get it now!
oci-dbaas .cf	USD 0.00	Get it now!
oci-dbaas .gq	USD 0.00	Get it now!

3. Click **Get it now!** and then click **Checkout**.
4. Put a temporary URL in the box and click **Continue**.



The screenshot shows the checkout page for the domain 'oci-dbaas.tk'. It features a 'Forward this domain' button and a 'Use DNS' button. Below these buttons, there is a text input field containing 'http://cloud.oracle.com'. A 'Continue' button is located at the bottom right of the form.

5. Select the check box for the agreement, and then click **Complete Order**.



The screenshot shows the agreement section of the checkout page. It includes a checked checkbox next to the text 'I have read and agree to the Terms & Conditions' and a 'Complete Order' button.

6. Navigate to **Services > My Domains** and click **Manage Domain** for the domain that you purchased.
7. Go to the **Manage Freenom DNS** tab, and then click **Edit Nameservers**.

8. Select **Use custom nameservers**, enter the following Oracle Cloud Infrastructure DNS name servers, and click **Change Nameservers**.
 - ns1.p68.dns.oraclecloud.net
 - ns2.p68.dns.oraclecloud.net
 - ns3.p68.dns.oraclecloud.net
 - ns4.p68.dns.oraclecloud.net

The screenshot shows the Oracle Cloud Infrastructure console interface for managing the domain 'oci-dbaas.tk'. The 'Nameservers' section is active, and the 'Use custom nameservers (enter below)' option is selected. The following nameservers are entered:

- Nameserver 1: ns1.p68.dns.oraclecloud.net
- Nameserver 2: ns2.p68.dns.oraclecloud.net
- Nameserver 3: ns3.p68.dns.oraclecloud.net
- Nameserver 4: ns4.p68.dns.oraclecloud.net
- Nameserver 5: (empty)

A 'Change Nameservers' button is located at the bottom of the form.

Step 4: Run the Terraform Script

When you run the script, it performs the following steps based on the options that you selected:

1. Creates a compute instance (of the shape that you selected) on the public subnet that you defined in the VCN
2. Downloads the relevant binaries to the compute VM: `ords.war`, APEX, JDK, Tomcat (if selected), and `acme.sh` (FQDN-based access only)
3. Installs the downloaded binaries
4. (IP-based access only) Configures SSL configuration by using self-signed certificates on the web container
5. Installs Oracle APEX on the target database
6. (FQDN-based access only) Configures the Oracle Cloud Infrastructure DNS service:
 - A. Creates a zone
 - B. Registers an A record for the compute instance and publishes the change
7. (FQDN-based access only) Acquires a Let's Encrypt certificate
8. Configures SSL configuration on the web container

In our testing, the script completed within 30 minutes for any combination of installation options.

Tip: If the script is running perpetually, verify that all the networking prerequisites are met, and then re-run the script.

Sample Output of Running the Script

```
$ terraform init
-- Resource Planning
$ terraform plan
=> Prompt is showing up for Target DB's admin password (default setting)
-- Resource Building
$ terraform apply
=> Prompt is showing up for Target DB's admin password (default setting)
```

Configuring Oracle REST Data Services to Access More Than One Database Instance

You can configure Oracle REST Data Services to access more than one database instance. The `apex_add_db.sh` shell script is bundled with the Terraform script. You must run this script with the following parameters to configure multiple databases with the same Oracle REST Data Services installation. This script must be run on the Oracle REST Data Services compute host by the `oracle` user (when using a Jetty server) or the `tomcat` user (when using Tomcat).

Run this script once for each additional database that you want to configure.

```
$ ./apex_add_db.sh -p <database_admin_password> -i <IP_address> -s <database_service_name>
```

Step 5: Access Oracle REST Data Services

The Oracle REST Data Services web browser URL follows this format:

```
https://<compute_IP_address> or <FQDN>:<port>/ords/<database_name>/
```

Workspace: Internal

User: ADMIN

Password: <SYS_or_database_admin_password>

Maintenance

If you want to patch or upgrade Oracle APEX later, we recommend the following steps:

1. Shut down the compute VM that has Oracle REST Data Services running on it.
2. Re-run the Terraform template with latest Oracle APEX distribution zip file.
3. After you confirm that the installation on the new compute VM is successful, terminate the older VM.

If you want to remove the Oracle APEX installation, perform the following steps:

1. Run `@apxremov.sql` on the compute VM by connecting to the target database instance via SQL*Plus. This action removes the Oracle APEX installation.
2. Run the Terraform script using the Terraform Destroy command. This command terminates the Oracle REST Data Services compute instance. Following is sample output of running the script to destroy created resources:

```
### -- Resource Termination
$ terraform destroy
```

Conclusion

This document describes the process of installing Oracle APEX on Oracle Cloud Infrastructure Database using the Terraform template released by Oracle. This template greatly simplifies the installation process while implementing best practices for Oracle APEX deployment.

At a high level, the Terraform template performs the following actions:

- Creates a compute VM in a public subnet and configures Oracle REST Data Services on it
- Creates and configures the Oracle APEX database schema
- Performs SSL configuration using either self-signed certificates or Let's Encrypt certificates on the web container

This architecture improves security by not exposing the database to the internet, reduces resource overhead by allowing the same Oracle REST Data Services instance to access multiple Oracle APEX databases, and greatly simplifies Oracle REST Data Services maintenance tasks by separating Oracle REST Data Services from the APEX installation.

References

- [Let's Encrypt! Nginx on Oracle Cloud Infrastructure Using Oracle Linux](#)
- [Using Let's Encrypt with Oracle Linux in Oracle Cloud Infrastructure](#)
- [Oracle APEX documentation](#)

Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120
