

# Oracle Things

- [APEX on Docker](#)
- [OCI CLI setup](#)
- [Oracle Database 19c EE](#)
- [Oracle Database 21c](#)
- [Oracle Enterprise Manager 13.5 on OL8](#)
- [Oracle Enterprise Manager Cloud Control 13c](#)
- [Oracle ORDS and Apache](#)
- [Oracle Response Files](#)
- [Oracle XE and APEX on CentOS 7](#)

# APEX on Docker

```
docker network create ora
```

```
docker run --name oracledb \  
--network=ora \  
-p 1521:1521 \  
-p 5500:5500 \  
-v ~/oradata:/opt/oracle/oradata \  
-v ~/apex/images/apex:/tmp/apex_install \  
-e TZ=America/New_York \oracle/database:18.3.0-se2
```

```
docker exec oracledb ./setPassword.sh Oradoc_db1
```

```
# Install and configure APEXdocker exec -it oracledb bash -c "source /home/oracle/.bashrc; bash"
```

```
cd /tmp/apex-installsqlplus sys/Oradoc_db1@localhost/orclpdb1 as sysdba
```

```
-- Install APEX@apexins.sql SYSAUX SYSAUX TEMP /i/
```

```
-- APEX REST configuration@apex_rest_config_core.sql oracle oracle
```

```
alter user apex_public_user identified by oracle account unlock;
```

-- From the blog: "Create a network ACE for APEX (this is used when consuming Web services or sending outbound mail):"

```
declare  
l_acl_path varchar2(4000);  
l_apex_schema varchar2(100);  
begin  
for c1 in (select schema  
from sys.dba_registry  
where comp_id = 'APEX') loop  
l_apex_schema := c1.schema;  
end loop;  
sys.dbms_network_acl_admin.append_host_ace(
```

```
host => '*',
ace => xs$ace_type(privilege_list => xs$name_list('connect'),
principal_name => l_apex_schema,
principal_type => xs_acl.ptype_db));
commit;
end;
/
```

```
begin
apex_util.set_security_group_id( 10 );
apex_util.create_user(
p_user_name => 'ADMIN',
p_email_address => 'systems@example.com',
p_web_password => 'oracle',
p_developer_privs => 'ADMIN' );
apex_util.set_security_group_id( null );
commit;
end;
/
```

```
-- Exit SQLexit
```

## #### ORDS

-- Assuming that you have a folder called ~/docker/ordscd ~/docker/ords

```
git clone https://github.com/martindsouza/docker-ords.git
cd docker-ords
```

```
-- Extract the ords.war file from the ords.zip downloadunzip ~/Downloads/ords*.zip ords.war
```

```
cd ~/ords/docker-ords
ORDS_VERSION=18.3.0
docker build -t ords:$ORDS_VERSION .
```

```
docker run --name ords\
--network=ora \
```

```
-e TZ=America/New_York \  
-e DB_HOSTNAME=oracledb \  
-e DATABASE_SERVICE_NAME="orclpdb1" \  
-e DB_PORT=1521 \  
-e APEX_PUBLIC_USER_PASS=oracle \  
-e APEX_LISTENER_PASS=oracle \  
-e APEX_REST_PASS=oracle \  
-e ORDS_PASS=oracle \  
-e SYS_PASS=Oradoc_db1 \  
--volume ~/ords/ords-18.3.0/config:/opt/ords \  
--volume ~/apex/images/apex/images:/ords/apex-images \  
-p 8080:8080 \  
ords:18.3.0
```

```
docker run -t -i \  
--name ords \  
--network=ora \  
-e DATABASE_HOSTNAME="oracledb" \  
-e DATABASE_PORT="1521" \  
-e DATABASE_SERVICE_NAME="ORCLPDB1" \  
-e DATABASE_PUBLIC_USER_PASS=oracle \  
-e APEX_LISTENER_PASS=oracle \  
-e APEX_REST_PASS=oracle \  
-e ORDS_PASS=oracle \  
--volume ~/apex/images:/usr/local/tomcat/webapps/i \  
-p 8080:8080 \  
lucassampsouza/ords_apex:3.0.9
```

# OCI CLI setup

## Setup OCI

You'll need a terminal and a browser. Log in to OCI and go to your profile. You'll need the user OCID and the tenant OCID. Note the region you are in.

## Install oci

- Oracle Linux 8

```
sudo dnf -y install oraclelinux-developer-release-el8  
sudo dnf install python36-oci-cli
```

- Other Linux

```
bash -c "$(curl -L https://raw.githubusercontent.com/oracle/oci-cli/master/scripts/install/install.sh)"
```

- MacOS

```
brew upgrade && brew update && brew install oci-cli
```

- Windows

lol. No. Get a Linux box or use WSL.

## Configure

At the command line:

```
oci setup config
```

In a browser open the OCI console copy the user OCID.

Open the Profile menu (User menu icon) and click User Settings.

Copy the user OCID and return to the command line. Paste the user OCID at the User OCID prompt and hit return.

Return to the OCI console and navigate to Administration > Tenancy Details. In the Tenancy Information tab, click Copy to copy the OCID.

Return to the command line and paste the value at the OICD prompt and hit return.

Enter the associated region, hit enter.

Generate or enter the path to the private PEM file. For example, `~/.oci/oci_api_key.pem` If required, enter the passphrase used with the key. Enter y or no to the prompt asking about whether to store the passphrase.

The configuration process is complete.

## CLI Concepts

# Oracle Database 19c EE

Download the following software:

- [Oracle Linux 7 \(x86\\_64\)](#)
- [Oracle Database \(x86\\_64\) 19c Enterprise Edition](#)
- [Oracle APEX](#)

## OS setup

Install Oracle Linux 7. Select Server with a GUI. or Minimal.

As the root user, configure the OS and create the directory structure.

```
yum install -y oracle-database-preinstall-19c wget
mkdir -p /u01/app/oracle/product/19.0.0/dbhome_1
mkdir -p /u02/oradata
chown -R oracle:oinstall /u01 /u02
chmod -R 775 /u01 /u02
```

## Database installation.

**As the oracle user:**

Make a scripts folder

```
mkdir /home/oracle/scripts
```

Create an environment script. This will hold all of the settings.

```
cat > /home/oracle/scripts/setEnv.sh <<EOF
# Oracle Settings
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=ora19c.core.example.com
export ORACLE_UNQNAME=cdb1
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/19.0.0/dbhome_1
```

```
export ORA_INVENTORY=/u01/app/oralInventory
export ORACLE_SID=cdb1
export PDB_NAME=pdb1
export DATA_DIR=/u02/oradata
export PATH=/usr/sbin:/usr/local/bin:\$PATH
export PATH=\$ORACLE_HOME/bin:\$PATH
export LD_LIBRARY_PATH=\$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=\$ORACLE_HOME/jlib:\$ORACLE_HOME/rdbms/jlib
EOF
```

Add the contents of setEnv.sh to Oracle's .bash\_profile.

```
echo ". /home/oracle/scripts/setEnv.sh" >> /home/oracle/.bash_profile
```

Create a start script.

```
cat > /home/oracle/scripts/start_all.sh <<EOF
#!/bin/bash
. /home/oracle/scripts/setEnv.sh

export ORAENV_ASK=NO
. oraenv
export ORAENV_ASK=YES

dbstart \$ORACLE_HOME
EOF
```

Create a stop script.

```
cat > /home/oracle/scripts/stop_all.sh <<EOF
#!/bin/bash
. /home/oracle/scripts/setEnv.sh

export ORAENV_ASK=NO
. oraenv
export ORAENV_ASK=YES

dbshut \$ORACLE_HOME
EOF
```

Set the owner and folder and execute on the scripts.



```
chown -R oracle:oinstall /home/oracle/scripts
chmod u+x /home/oracle/scripts/*.sh
```

Load the environment.

```
source /home/oracle/.bash_profile
```

Unzip the installer.

```
cd $ORACLE_HOME
unzip -oq /tmp/LINUX.X64_193000_db_home.zip
```

Run the installer.

```
./runInstaller -ignorePrereq -waitforcompletion -silent \
  -responseFile ${ORACLE_HOME}/install/response/db_install.rsp \
  oracle.install.option=INSTALL_DB_SWONLY \
  ORACLE_HOSTNAME=${ORACLE_HOSTNAME} \
  UNIX_GROUP_NAME=oinstall \
  INVENTORY_LOCATION=${ORA_INVENTORY} \
  SELECTED_LANGUAGES=en,en_US \
  ORACLE_HOME=${ORACLE_HOME} \
  ORACLE_BASE=${ORACLE_BASE} \
  oracle.install.db.InstallEdition=EE \
  oracle.install.db.OSDBA_GROUP=dba \
  oracle.install.db.OSBACKUPDBA_GROUP=dba \
  oracle.install.db.OSDGDBA_GROUP=dba \
  oracle.install.db.OSKMDBA_GROUP=dba \
  oracle.install.db.OSRACDBA_GROUP=dba \
  SECURITY_UPDATES_VIA_MYORACLESUPPORT=false \
  DECLINE_SECURITY_UPDATES=true
```

As a **root** user, execute the following:

```
/u01/app/oraInventory/orainstRoot.sh
/u01/app/oracle/product/19.0.0/dbhome_1/root.sh
```

As the **oracle** user:

Start the listener

```
lsnrctl start
```

Create the first database.

```
dbca -silent -createDatabase \
  -templateName General_Purpose.dbc \
  -gdbname ${ORACLE_SID} -sid ${ORACLE_SID} -responseFile NO_VALUE \
  -characterSet AL32UTF8 \
  -sysPassword Som3bTt3rpwd \
  -systemPassword Som3bTt3rpwd \
  -createAsContainerDatabase true \
  -numberOfPDBs 1 \
  -pdbName ${PDB_NAME} \
  -pdbAdminPassword Som3bTt3rpwd \
  -databaseType MULTIPURPOSE \
  -automaticMemoryManagement false \
  -totalMemory 2000 \
  -storageType FS \
  -datafileDestination "${DATA_DIR}" \
  -redoLogFileSize 50 \
  -emConfiguration NONE \
  -ignorePreReqs
```

Edit the "/etc/oratab" file setting the restart flag for each instance to 'Y'.

Example:

```
cdb1:/u01/app/oracle/product/19.0.0/db_1:Y
```

Enable Oracle Managed Files and make the PDB start when the instance starts

```
sqlplus / as sysdba <<EOF
alter system set db_create_file_dest='/u02/oradata';
alter pluggable database pdb1 save state;
exit;
EOF
```

# APEX

As the **oracle** user:

Make the apex directory and unzip the apex files.

```
mkdir -p /home/oracle/apex
unzip /tmp/apex_20.*.zip -d /home/oracle
chown -R oracle:oinstall /home/oracle/apex
cd /home/oracle/apex
```

Create an ACL script. This will be needed later.

```
cat > apex_acl.sql << EOF
BEGIN
  BEGIN
    dbms_network_acl_admin.drop_acl(acl => 'all-network-PUBLIC.xml');
  EXCEPTION
    WHEN OTHERS THEN
      NULL;
  END;
  dbms_network_acl_admin.create_acl(acl      => 'all-network-PUBLIC.xml',
                                   description => 'Allow all network traffic',
                                   principal  => 'PUBLIC',
                                   is_grant   => TRUE,
                                   privilege  => 'connect');
  dbms_network_acl_admin.add_privilege(acl      => 'all-network-PUBLIC.xml',
                                       principal => 'PUBLIC',
                                       is_grant  => TRUE,
                                       privilege => 'resolve');
  dbms_network_acl_admin.assign_acl(acl => 'all-network-PUBLIC.xml',
                                   host => '*');

END;
/
sho err
COMMIT;
/
EOF
```

Connect to the database

```
sqlplus /nolog
```

Change roles

```
CONN pdb1 AS SYSDBA
```

```
alter session set container=PDB1;
```

Run the script to install a full development environment

```
@apexins.sql SYSAUX SYSAUX TEMP /i/
```

Create an instance administrator user and set their password

```
@apxchpwd.sql
```

Configure REST Data Services

```
@apex_rest_config.sql
```

Run the ACL setup script created earlier.

```
@apex_acl.sql
```

Unlock APEX users

```
ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK;  
ALTER USER APEX_PUBLIC_USER IDENTIFIED BY "Som3bTt3rpwd";  
ALTER USER APEX_REST_PUBLIC_USER IDENTIFIED BY "Som3bTt3rpwd" ACCOUNT UNLOCK;  
ALTER USER APEX_LISTENER IDENTIFIED BY "Som3bTt3rpwd" ACCOUNT UNLOCK;
```

# Oracle Database 21c

Download the following software:

- [Oracle Linux 8 \(x86\\_64\)](#)
- [Oracle Database 21c Enterprise Edition \(x86\\_64\)](#)

## NOTICE

This is a very basic install. It should only be used as a guide.

## OS setup

Install Oracle Linux 8. "Minimal" installation.

As the root user, configure the OS and create the required directory structure.

```
dnf install -y oracle-database-preinstall-21c wget
mkdir -p /u01/app/oracle/product/21.0.0/dbhome_1
mkdir -p /u02/oradata
chown -R oracle:oinstall /u01 /u02
chmod -R 775 /u01 /u02
```

## Database installation

**As the oracle user:**

Create a scripts folder:

```
mkdir /home/oracle/scripts
```

Create an environment script. This will hold all the necessary settings.

```
cat > /home/oracle/scripts/setEnv.sh <<EOF
# Oracle Settings
export TMP=/tmp
export TMPDIR=\$TMP
export ORACLE_HOSTNAME=ora21c.example.com
export ORACLE_UNQNAME=cdb1
```

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=\$ORACLE_BASE/product/21.0.0/dbhome_1
export ORA_INVENTORY=/u01/app/oraInventory
export ORACLE_SID=cdb1
export PDB_NAME=pdb1
export DATA_DIR=/u02/oradata
export PATH=/usr/sbin:/usr/local/bin:\$PATH
export PATH=\$ORACLE_HOME/bin:\$PATH
export LD_LIBRARY_PATH=\$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=\$ORACLE_HOME/jlib:\$ORACLE_HOME/rdbms/jlib
EOF
```

Add the contents of `setEnv.sh` to Oracle's `.bash_profile`.

```
echo ". /home/oracle/scripts/setEnv.sh" >> /home/oracle/.bash_profile
```

Create a start script.

```
cat > /home/oracle/scripts/start_all.sh <<EOF
#!/bin/bash
. /home/oracle/scripts/setEnv.sh

export ORAENV_ASK=NO
. oraenv
export ORAENV_ASK=YES

dbstart \$ORACLE_HOME
EOF
```

Create a stop script.

```
cat > /home/oracle/scripts/stop_all.sh <<EOF
#!/bin/bash
. /home/oracle/scripts/setEnv.sh

export ORAENV_ASK=NO
. oraenv
export ORAENV_ASK=YES

dbshut \$ORACLE_HOME
EOF
```

Set the owner and permissions for the scripts.

```
chown -R oracle:oinstall /home/oracle/scripts
chmod u+x /home/oracle/scripts/*.sh
```

Load the environment.

```
source /home/oracle/.bash_profile
```

Unzip the database installer.

```
cd $ORACLE_HOME
unzip -oq /tmp/LINUX.X64_210000_db_home.zip
```

Run the installer.

```
./runInstaller -ignorePrereq -waitforcompletion -silent \
  -responseFile ${ORACLE_HOME}/install/response/db_install.rsp \
  oracle.install.option=INSTALL_DB_SWONLY \
  ORACLE_HOSTNAME=${ORACLE_HOSTNAME} \
  UNIX_GROUP_NAME=oinstall \
  INVENTORY_LOCATION=${ORA_INVENTORY} \
  SELECTED_LANGUAGES=en,en_US \
  ORACLE_HOME=${ORACLE_HOME} \
  ORACLE_BASE=${ORACLE_BASE} \
  oracle.install.db.InstallEdition=EE \
  oracle.install.db.OSDBA_GROUP=dba \
  oracle.install.db.OSBACKUPDBA_GROUP=dba \
  oracle.install.db.OSDGDBA_GROUP=dba \
  oracle.install.db.OSKMDBA_GROUP=dba \
  oracle.install.db.OSRACDBA_GROUP=dba \
  SECURITY_UPDATES_VIA_MYORACLESUPPORT=false \
  DECLINE_SECURITY_UPDATES=true
```

As a **root** user, execute the following:

```
/u01/app/oraInventory/orainstRoot.sh
/u01/app/oracle/product/21.0.0/dbhome_1/root.sh
```

As the **oracle** user:

Start the listener.

```
lsnrctl start
```

Create the database.

```
dbca -silent -createDatabase \
  -templateName General_Purpose.dbc \
  -gdbname ${ORACLE_SID} -sid ${ORACLE_SID} -responseFile NO_VALUE \
  -characterSet AL32UTF8 \
  -sysPassword Som3bTt3rpwd \
  -systemPassword Som3bTt3rpwd \
  -createAsContainerDatabase true \
  -numberOfPDBs 1 \
  -pdbName ${PDB_NAME} \
  -pdbAdminPassword Som3bTt3rpwd \
  -databaseType MULTIPURPOSE \
  -automaticMemoryManagement false \
  -totalMemory 2000 \
  -storageType FS \
  -datafileDestination "${DATA_DIR}" \
  -redoLogFileSize 50 \
  -emConfiguration NONE \
  -ignorePreReqs
```

Edit the `/etc/oratab` file setting the restart flag for each instance to 'Y'.

Example:

```
cdb1:/u01/app/oracle/product/21.0.0/db_1:Y
```

Enable Oracle Managed Files and make the PDB start when the instance starts.

```
sqlplus / as sysdba <<EOF
alter system set db_create_file_dest='/u02/oradata';
alter pluggable database pdb1 save state;
exit;
EOF
```



# Oracle Enterprise Manager 13.5 on OL8

Oracle Enterprise Manager Cloud Control 13c on Oracle Linux 8

## Step 1: Download Required Software

First, ensure you have downloaded the necessary software from Oracle's official website:

- Oracle Linux 8 ISO from [Oracle Linux Download](#)
- Oracle Database 19c (or latest) from [Oracle Database Downloads](#)
- Oracle Enterprise Manager Cloud Control 13c Release 5 (or latest) from [Enterprise Manager Downloads](#)

## Step 2: Install Oracle Linux 8

1. **Install Oracle Linux 8:** Boot from the Oracle Linux 8 ISO and follow the installation prompts. If a GUI is needed, make sure to select the "Workstation" or "Server with GUI" base environment during installation.
2. **Set Up Network and Hostname:** Configure your network settings and hostname during or after installation as required.

## Step 3: Prepare the Operating System

1. **Update the System:** Ensure your system is up-to-date.

```
dnf update -y
```

2. **Install Required Packages:** The `oracle-database-preinstall-19c` package will automatically install dependencies and configure system parameters.

```
dnf install -y oracle-database-preinstall-19c wget
```

3. **Additional Dependencies:** If there are specific additional dependencies for Oracle Database 19c or the Enterprise Manager, install them as needed.

# Step 4: Configure System Settings and Users

1. **Directory Structure:** Create directories for Oracle software and data files.

```
mkdir -p /u01/app/oracle/product/19.0.0/dbhome_1
mkdir -p /u01/app/oracle/middleware
mkdir -p /u01/app/oracle/agent
mkdir -p /u01/tmp
mkdir -p /u01/oradata
chown -R oracle:oinstall /u01
chmod -R 775 /u01
```

2. **Environment Variables:** As the `oracle` user, configure environment variables.

- Create a script `/home/oracle/scripts/setEnv.sh` with the following content:

```
export TMP=/tmp
export TMPDIR=$TMP

export ORACLE_HOSTNAME=<your_hostname>
export ORACLE_UNQNAME=emcdb
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/19.0.0/dbhome_1
export ORACLE_SID=emcdb

export PATH=/usr/sbin:/usr/local/bin:$PATH
export PATH=$ORACLE_HOME/bin:$PATH

export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib

export OMS_HOME=/u01/app/oracle/middleware
export AGENT_HOME=/u01/app/oracle/agent/agent_inst

export MW_HOME=${ORACLE_BASE}/middleware
export OMS_HOME=${MW_HOME}
export GC_INST=${ORACLE_BASE}/gc_inst
export AGENT_BASE=${ORACLE_BASE}/agent
```

- Add the script to the oracle user's `.bash_profile`.

```
echo ". /home/oracle/scripts/setEnv.sh" >> /home/oracle/.bash_profile
```

## Step 5: Install Oracle Database

1. **Prepare the Database Installation:** Unzip the Oracle Database software and prepare a response file for a silent installation.

```
mkdir /u01/software/  
cd /u01/software  
unzip <path_to_your_downloaded_db_software>.zip  
cd database
```

2. **Run the Database Installer:** Execute the silent installation using parameters.

```
./runInstaller -ignorePrereq -waitforcompletion -silent \
-responseFile ${ORACLE_HOME}/install/response/db_install.rsp \
oracle.install.option=INSTALL_DB_SWONLY \
ORACLE_HOSTNAME=${ORACLE_HOSTNAME} \
UNIX_GROUP_NAME=oinstall \
INVENTORY_LOCATION=${ORA_INVENTORY} \
SELECTED_LANGUAGES=en,en_US \
ORACLE_HOME=${ORACLE_HOME} \
ORACLE_BASE=${ORACLE_BASE} \
oracle.install.db.InstallEdition=EE \
oracle.install.db.OSDBA_GROUP=dba \
oracle.install.db.OSBACKUPDBA_GROUP=dba \
oracle.install.db.OSDGDBA_GROUP=dba \
oracle.install.db.OSKMDBA_GROUP=dba \
oracle.install.db.OSRACDBA_GROUP=dba \
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false \
DECLINE_SECURITY_UPDATES=true
```

3. **Configure the Listener and Database:** Use the `netca` and `dbca` tools to configure the network and create the database, respectively. These can be automated with response files or command-line options for silent operations.

Start the listener:

```
lsnrctl start
```

Create the database:

```
dbca -silent -createDatabase \
    -templateName General_Purpose.dbc \
    -gdbname ${ORACLE_SID} -sid ${ORACLE_SID} -responseFile NO_VALUE \
    -characterSet AL32UTF8 \
    -sysPassword Som3bTt3rpwd \
    -systemPassword Som3bTt3rpwd \
    -createAsContainerDatabase true \
    -numberOfPDBs 1 \
    -pdbName ${PDB_NAME} \
    -pdbAdminPassword Som3bTt3rpwd \
    -databaseType MULTIPURPOSE \
    -automaticMemoryManagement false \
    -totalMemory 8000 \
    -storageType FS \
    -datafileDestination "${DATA_DIR}" \
    -redoLogFileSize 50 \
    -emConfiguration NONE \
    -ignorePreReqs
```

#### 4. Set the pluggable database to auto-start.

```
sqlplus / as sysdba <<EOF
alter system set db_create_file_dest='/u01/oradata';
alter pluggable database emrep save state;

-- Recommended settings
alter system set "_allow_insert_with_update_check"=true scope=both;
alter system set session_cached_cursors=200 scope=spfile;
alter system set sga_target=800M scope=both;
alter system set pga_aggregate_target=450M scope=both;

EOF
```

## Step 6: Install Oracle Enterprise Manager Cloud Control

### 1. **Prepare for Installation:** Unzip

the Enterprise Manager software to `/u01/software/em` and navigate to the directory.

2. **Enterprise Manager Install Response File:** Similar to the database, prepare a response file for the Enterprise Manager installation.

Setup the environment:

```
export ORA_INVENTORY=/u01/app/oraInventory
export PDB_NAME=emrep
export SYS_PASSWORD=SysAdminpW1
export UNIX_GROUP_NAME=oinstall
export MW_HOME=${ORACLE_BASE}/middleware
export OMS_HOME=${MW_HOME}
export GC_INST=${ORACLE_BASE}/gc_inst
export AGENT_BASE=${ORACLE_BASE}/agent
export WLS_USERNAME=weblogic
export WLS_PASSWORD=SysAdminpW1
export SYSMAN_PASSWORD=${WLS_PASSWORD}
export AGENT_PASSWORD=${WLS_PASSWORD}
export SOFTWARE_LIBRARY=${ORACLE_BASE}/swlib
export DATABASE_HOSTNAME=localhost
export LISTENER_PORT=1521
export SOFTWARE_DIR=/u01/software/em
```

Create the response file:

```
cat > /tmp/install.rsp <<EOF
RESPONSEFILE_VERSION=2.2.1.0.0
UNIX_GROUP_NAME=${UNIX_GROUP_NAME}
INVENTORY_LOCATION=${ORA_INVENTORY}
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
DECLINE_SECURITY_UPDATES=true
INSTALL_UPDATES_SELECTION=skip
ORACLE_MIDDLEWARE_HOME_LOCATION=${MW_HOME}
ORACLE_HOSTNAME=${ORACLE_HOSTNAME}
AGENT_BASE_DIR=${AGENT_BASE}
WLS_ADMIN_SERVER_USERNAME=${WLS_USERNAME}
WLS_ADMIN_SERVER_PASSWORD=${WLS_PASSWORD}
WLS_ADMIN_SERVER_CONFIRM_PASSWORD=${WLS_PASSWORD}
NODE_MANAGER_PASSWORD=${WLS_PASSWORD}
NODE_MANAGER_CONFIRM_PASSWORD=${WLS_PASSWORD}
ORACLE_INSTANCE_HOME_LOCATION=${GC_INST}
```

```
CONFIGURE_ORACLE_SOFTWARE_LIBRARY=true
SOFTWARE_LIBRARY_LOCATION=${SOFTWARE_LIBRARY}
DATABASE_HOSTNAME=${DATABASE_HOSTNAME}
LISTENER_PORT=${LISTENER_PORT}
SERVICENAME_OR_SID=${PDB_NAME}
SYS_PASSWORD=${SYS_PASSWORD}
SYSMAN_PASSWORD=${SYSMAN_PASSWORD}
SYSMAN_CONFIRM_PASSWORD=${SYSMAN_PASSWORD}
DEPLOYMENT_SIZE=SMALL
AGENT_REGISTRATION_PASSWORD=${AGENT_PASSWORD}
AGENT_REGISTRATION_CONFIRM_PASSWORD=${AGENT_PASSWORD}
PLUGIN_SELECTION={}
b_upgrade=false
EM_INSTALL_TYPE=NOSEED
CONFIGURATION_TYPE=LATER
CONFIGURE_SHARED_LOCATION_BIP=false
EOF
```

3. **Run the Installer:** Execute the silent installation using the response file.

```
./em13500_linux64.bin silent -responseFile /tmp/install.rsp -J-Djava.io.tmpdir=/u01/tmp/
```

4. **Post-Installation Steps:**

At the end of a successful install, run the root script.

**As root**

```
sh ${MW_HOME}/allroot.sh
```

**Return to Oracle user**

Create the the configuration response file..

```
cat > /tmp/config.rsp <<EOF
RESPONSEFILE_VERSION=2.2.1.0.0
UNIX_GROUP_NAME=${UNIX_GROUP_NAME}
INVENTORY_LOCATION=${ORA_INVENTORY}
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
DECLINE_SECURITY_UPDATES=true
INSTALL_UPDATES_SELECTION=skip
ORACLE_MIDDLEWARE_HOME_LOCATION=${MW_HOME}
ORACLE_HOSTNAME=${ORACLE_HOSTNAME}
AGENT_BASE_DIR=${AGENT_BASE}
```

```
WLS_ADMIN_SERVER_USERNAME=${WLS_USERNAME}
WLS_ADMIN_SERVER_PASSWORD=${WLS_PASSWORD}
WLS_ADMIN_SERVER_CONFIRM_PASSWORD=${WLS_PASSWORD}
NODE_MANAGER_PASSWORD=${WLS_PASSWORD}
NODE_MANAGER_CONFIRM_PASSWORD=${WLS_PASSWORD}
ORACLE_INSTANCE_HOME_LOCATION=${GC_INST}
CONFIGURE_ORACLE_SOFTWARE_LIBRARY=true
SOFTWARE_LIBRARY_LOCATION=${SOFTWARE_LIBRARY}
DATABASE_HOSTNAME=${DATABASE_HOSTNAME}
LISTENER_PORT=${LISTENER_PORT}
SERVICENAME_OR_SID=${PDB_NAME}
SYS_PASSWORD=${SYS_PASSWORD}
SYSMAN_PASSWORD=${SYSMAN_PASSWORD}
SYSMAN_CONFIRM_PASSWORD=${SYSMAN_PASSWORD}
DEPLOYMENT_SIZE=SMALL
AGENT_REGISTRATION_PASSWORD=${AGENT_PASSWORD}
AGENT_REGISTRATION_CONFIRM_PASSWORD=${AGENT_PASSWORD}
PLUGIN_SELECTION={}
b_upgrade=false
EM_INSTALL_TYPE=NOSEED
CONFIGURATION_TYPE=ADVANCED
CONFIGURE_SHARED_LOCATION_BIP=false
EOF
```

Run the configuration. This will take a very long time.

```
${MW_HOME}/sysman/install/ConfigureGC.sh -silent -responseFile /tmp/config.rsp
```

At the end of the installation you'll be provided a list of URLs and post-installation steps. Follow any post-installation steps such as deploying agents, as described in the Enterprise Manager documentation.

## Step 7: Finalize Installation

- **Start Services:** Ensure that the Oracle database and the Enterprise Manager services are started.
- **Verify Installation:** Access the Enterprise Manager web interface to confirm that the installation was successful.

# Notes:

- **Response Files:** The details on what to include in response files for both the database and Enterprise Manager can be found in the official Oracle documentation. These files control installation options, such as directories, components to install, and initial setup parameters.
- **Oracle Documentation:** For detailed options for the `dbca`, `netca`, and Enterprise Manager silent installation parameters, refer to Oracle's official documentation.

This guide outlines a general approach to installing Oracle Database and Oracle Enterprise Manager Cloud Control on Oracle Linux 8. Tailor the response files to your environment's specific needs, referring to Oracle's documentation for the most accurate and detailed instructions.



# Oracle Enterprise Manager Cloud Control 13c

Download the following software:

- [Oracle Linux 7 \(x86\\_64\)](#)
- [Oracle Database \(x86\\_64\) 12c, 18c or 19c Enterprise Edition](#)
- [Enterprise Manager Cloud Control 13c Release 3 \(13.3.0.0\) \(x86\\_64\)](#)

## OS Installation

Install Oracle Linux 7. Choose "Server with a GUI" during installation.

Do the following as root.

### 1. Install required packages

```
yum install oracle-database-server-12cR2-preinstall -y
yum install wget -y
yum install make -y
yum install binutils -y
yum install gcc -y
yum install libaio -y
yum install glibc-common -y
yum install libstdc++ -y
yum install sysstat -y
yum install glibc -y
yum install glibc-devel.i686 -y
yum install glibc-devel -y
yum install libXtst -y
```

### 2. Create the directory structures

```
mkdir -p /u01/app/oracle/product/12.2.0.1/db_1
mkdir -p /u01/app/oracle/middleware
mkdir -p /u01/app/oracle/agent
```

```
mkdir -p /u01/tmp
mkdir -p /u01/oradata
chown -R oracle:oinstall /u01
chmod -R 775 /u01
```

## Database Installation

Do the following as the oracle user

1. Create a scripts directory.

```
mkdir /home/oracle/scripts
```

2. Create `/home/oracle/scripts/setEnv.sh` with the following contents:

```
export TMP=/tmp
export TMPDIR=$TMP

export ORACLE_HOSTNAME=oraem.servers-farm.io
export ORACLE_UNQNAME=emcdb
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/12.2.0.1/db_1
export ORACLE_SID=emcdb

export PATH=/usr/sbin:/usr/local/bin:$PATH
export PATH=$ORACLE_HOME/bin:$PATH

export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib

export OMS_HOME=/u01/app/oracle/middleware
export AGENT_HOME=/u01/app/oracle/agent/agent_inst

export MW_HOME=${ORACLE_BASE}/middleware
export OMS_HOME=${MW_HOME}
export GC_INST=${ORACLE_BASE}/gc_inst
export AGENT_BASE=${ORACLE_BASE}/agent
```

3. Add the new script to your bash profile.

```
echo ". /home/oracle/scripts/setEnv.sh" >> /home/oracle/.bash_profile
```

#### 4. Create a software folder and unzip the database.zip download in it.

```
mkdir /u01/software/  
cd /u01/software  
unzip ~/odb12c.zip  
cd database
```

#### 5. Run the silent database install.

```
./runInstaller -ignorePrereq -waitforcompletion -silent \\  
-responseFile /u01/software/database/response/db_install.rsp \\  
oracle.install.option=INSTALL_DB_SWONLY \\  
ORACLE_HOSTNAME=${HOSTNAME} \\  
UNIX_GROUP_NAME=oinstall \\  
INVENTORY_LOCATION=/u01/app/oraInventory \\  
SELECTED_LANGUAGES=en,en_US \\  
ORACLE_HOME=${ORACLE_HOME} \\  
ORACLE_BASE=${ORACLE_BASE} \\  
oracle.install.db.InstallEdition=EE \\  
oracle.install.db.OSDBA_GROUP=dba \\  
oracle.install.db.OSBACKUPDBA_GROUP=dba \\  
oracle.install.db.OSDGDBA_GROUP=dba \\  
oracle.install.db.OSKMDBA_GROUP=dba \\  
oracle.install.db.OSRACDBA_GROUP=dba \\  
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false \\  
DECLINE_SECURITY_UPDATES=true
```

#### 6. Start the listener

```
lsnrctl start
```

#### 7. Create the repository database

```
dbca -silent -createDatabase \\  
-templateName General_Purpose.dbc \\  
-gdbname emcdb -sid emcdb -responseFile NO_VALUE \\  
-characterSet AL32UTF8 \\  
-sysPassword SysAdminpW1 \\  
-systemPassword SysAdminpW1 \\  
-createAsContainerDatabase true \\  
-numberOfPDBs 1 \
```

```
-pdbName emrep \
-pdbAdminPassword SysAdminpW1 \
-databaseType MULTIPURPOSE \
-automaticMemoryManagement false \
-totalMemory 2000 \
-storageType FS \
-datafileDestination /u01/oradata \
-redoLogFileSize 50 \
-emConfiguration NONE \
-ignorePreReqs
```

## 8. Set the pluggable database to auto-start.

```
sqlplus / as sysdba <<EOF
alter system set db_create_file_dest='/u01/oradata';
alter pluggable database emrep save state;

-- Recommended settings
alter system set "_allow_insert_with_update_check"=true scope=both;
alter system set session_cached_cursors=200 scope=spfile;
alter system set sga_target=800M scope=both;
alter system set pga_aggregate_target=450M scope=both;

-- For 12.1.0.2
alter system set optimizer_adaptive_features=false scope=both;
exit;
EOF
```

# Enterprise Manager Installation

Do the following as the oracle user.

1. Follow the extraction guide downloaded with the EM files. Place all of the files in /u01/software/em.

2, Setup the environment

```
export ORA_INVENTORY=/u01/app/oraInventory
export PDB_NAME=emrep
export SYS_PASSWORD=SysAdminpW1
```

```
export UNIX_GROUP_NAME=oinstall
export MW_HOME=${ORACLE_BASE}/middleware
export OMS_HOME=${MW_HOME}
export GC_INST=${ORACLE_BASE}/gc_inst
export AGENT_BASE=${ORACLE_BASE}/agent
export WLS_USERNAME=weblogic
export WLS_PASSWORD=SysAdminpw1
export SYSMAN_PASSWORD=${WLS_PASSWORD}
export AGENT_PASSWORD=${WLS_PASSWORD}
export SOFTWARE_LIBRARY=${ORACLE_BASE}/swlib
export DATABASE_HOSTNAME=localhost
export LISTENER_PORT=1521
export SOFTWARE_DIR=/u01/software/em
```

2. Change to the software directory and create a response file.

```
cat > /tmp/install.rsp <<EOF
RESPONSEFILE_VERSION=2.2.1.0.0
UNIX_GROUP_NAME=${UNIX_GROUP_NAME}
INVENTORY_LOCATION=${ORA_INVENTORY}
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
DECLINE_SECURITY_UPDATES=true
INSTALL_UPDATES_SELECTION=skip
ORACLE_MIDDLEWARE_HOME_LOCATION=${MW_HOME}
ORACLE_HOSTNAME=${ORACLE_HOSTNAME}
AGENT_BASE_DIR=${AGENT_BASE}
WLS_ADMIN_SERVER_USERNAME=${WLS_USERNAME}
WLS_ADMIN_SERVER_PASSWORD=${WLS_PASSWORD}
WLS_ADMIN_SERVER_CONFIRM_PASSWORD=${WLS_PASSWORD}
NODE_MANAGER_PASSWORD=${WLS_PASSWORD}
NODE_MANAGER_CONFIRM_PASSWORD=${WLS_PASSWORD}
ORACLE_INSTANCE_HOME_LOCATION=${GC_INST}
CONFIGURE_ORACLE_SOFTWARE_LIBRARY=true
SOFTWARE_LIBRARY_LOCATION=${SOFTWARE_LIBRARY}
DATABASE_HOSTNAME=${DATABASE_HOSTNAME}
LISTENER_PORT=${LISTENER_PORT}
SERVICENAME_OR_SID=${PDB_NAME}
SYS_PASSWORD=${SYS_PASSWORD}
SYSMAN_PASSWORD=${SYSMAN_PASSWORD}
SYSMAN_CONFIRM_PASSWORD=${SYSMAN_PASSWORD}
```

```
DEPLOYMENT_SIZE=SMALL
AGENT_REGISTRATION_PASSWORD=${AGENT_PASSWORD}
AGENT_REGISTRATION_CONFIRM_PASSWORD=${AGENT_PASSWORD}
PLUGIN_SELECTION={}
b_upgrade=false
EM_INSTALL_TYPE=NOSEED
CONFIGURATION_TYPE=LATER
CONFIGURE_SHARED_LOCATION_BIP=false
EOF
```

### 3. Run the installer.

```
./em13300_linux64.bin -silent -responseFile /tmp/install.rsp -J-Djava.io.tmpdir=/u01/tmp/
```

### 4. The the end of a successful install, run the root script.

```
sh ${MW_HOME}/allroot.sh
```

### 5. Create the the configuration response file..

```
cat > /tmp/config.rsp <<EOF
RESPONSEFILE_VERSION=2.2.1.0.0
UNIX_GROUP_NAME=${UNIX_GROUP_NAME}
INVENTORY_LOCATION=${ORA_INVENTORY}
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false
DECLINE_SECURITY_UPDATES=true
INSTALL_UPDATES_SELECTION=skip
ORACLE_MIDDLEWARE_HOME_LOCATION=${MW_HOME}
ORACLE_HOSTNAME=${ORACLE_HOSTNAME}
AGENT_BASE_DIR=${AGENT_BASE}
WLS_ADMIN_SERVER_USERNAME=${WLS_USERNAME}
WLS_ADMIN_SERVER_PASSWORD=${WLS_PASSWORD}
WLS_ADMIN_SERVER_CONFIRM_PASSWORD=${WLS_PASSWORD}
NODE_MANAGER_PASSWORD=${WLS_PASSWORD}
NODE_MANAGER_CONFIRM_PASSWORD=${WLS_PASSWORD}
ORACLE_INSTANCE_HOME_LOCATION=${GC_INST}
CONFIGURE_ORACLE_SOFTWARE_LIBRARY=true
SOFTWARE_LIBRARY_LOCATION=${SOFTWARE_LIBRARY}
DATABASE_HOSTNAME=${DATABASE_HOSTNAME}
LISTENER_PORT=${LISTENER_PORT}
SERVICENAME_OR_SID=${PDB_NAME}
```

```
SYS_PASSWORD=${SYS_PASSWORD}
SYSMAN_PASSWORD=${SYSMAN_PASSWORD}
SYSMAN_CONFIRM_PASSWORD=${SYSMAN_PASSWORD}
DEPLOYMENT_SIZE=SMALL
AGENT_REGISTRATION_PASSWORD=${AGENT_PASSWORD}
AGENT_REGISTRATION_CONFIRM_PASSWORD=${AGENT_PASSWORD}
PLUGIN_SELECTION={}
b_upgrade=false
EM_INSTALL_TYPE=NOSEED
CONFIGURATION_TYPE=ADVANCED
CONFIGURE_SHARED_LOCATION_BIP=false
EOF
```

6. Run the configuration. This will take a very long time.

```
${MW_HOME}/sysman/install/ConfigureGC.sh -silent -responseFile /tmp/config.rsp
```

At the end of the installation you'll be provided a list of URLs and post-installation steps.

# Oracle ORDS and Apache

Before using ORDS, you need a database and APEX. Go [here](#) to do that. Download [ORDS at Oracle](#).

## Installation of ORDS

The Oracle Rest Data Services (ORDS) installation consists of unzipping the downloaded archive, running the configuration command, then deploying. This setup is going to run ORDS in standalone mode with Apache HTTP to proxy requests.

As **oracle** user.

Download the ORDS zip to /tmp.

Extract the installer.

```
cd /tmp
mkdir -p /u01/ords
unzip ords-20.*.zip -d /u01/ords
```

Make a backup of the original properties file

```
mv /u01/ords/params/{ords_params.properties,ords_params.properties.orig}
```

Create a ords\_params.properties file.

```
cat > /u01/ords/params/ords_params.properties <<EOF
db.hostname=ora19c.core.example.com
db.port=1521
db.servicename=pdb1
db.sid=cdb1
db.username=APEX_PUBLIC_USER
db.password=Som3bTt3rpwd
migrate.apex.rest=false
plsql.gateway.add=true
rest.services.apex.add=true
rest.services.ords.add=true
schema.tablespace.default=SYSAUX
```



```
schema.tablespace.temp=TEMP
standalone.mode=true
standalone.use.https=true
standalone.http.port=8080
standalone.static.path=/home/oracle/apex/images
user.apex.listener.password=Som3bTt3rpwd
user.apex.restpublic.password=Som3bTt3rpwd
user.public.password=Som3bTt3rpwd
user.tablespace.default=APEX
user.tablespace.temp=TEMP
sys.user=SYS
sys.password=Som3bTt3rpwd
restEnabledSql.active=true
feature.sdw=true
database.api.enabled=true
EOF
```

Change to the ords directory and start the standalone instance.

```
cd /u01/ords
$JAVA_HOME/bin/java -jar ords.war standalone
```

## Configuration of Apache httpd to map ORDS

The last step is to configure Apache to map HTTP-requests to ORDS and therefore APEX engine.

For this, add a custom `httpd` configuration file. By default, every `.conf` file placed in the `etc/httpd/conf.d/` directory is read by `httpd` as an additional configuration file to the main `/etc/httpd/conf/httpd.conf` config file.

Create the `apex.conf` file in the `etc/httpd/conf.d/` directory with the contents as below:

```
# proxy ORDS
<VirtualHost *:80>
    # ServerName example.com
    # ServerAlias www.example.com

    # alias for APEX static files
```

```
Alias "/i" "/var/www/apex/images/"
```

```
# uncomment the line below if you want
```

```
# to redirect traffic to ORDS from root path
```

```
# RedirectMatch permanent "^/$" "/ords"
```

```
# proxy ORDS requests to tomcat
```

```
ProxyRequests off
```

```
<Location "/ords">
```

```
    ProxyPass "https://localhost:8443/ords"
```

```
    ProxyPassReverse "https://localhost:8443/ords"
```

```
</Location>
```

```
</VirtualHost>
```

Tell SELinux (Yes, that should be running) to allow Apache to communicate to ORDS.

```
setsebool httpd_can_network_connect on
```

# Oracle Response Files

## Oracle Database Install

Creating a response file for a silent installation of Oracle Database is a common task that allows you to automate the installation process without needing to interact with the graphical user interface. Below is an example response file for installing Oracle Database 19c on a Linux system. Keep in mind that response files can significantly vary depending on the Oracle Database version and the specific configurations you need. Adjustments may be required to fit your particular environment and requirements.

This example assumes you're installing Oracle Database 19c with a typical installation type, which includes the creation of a general-purpose database. You'll need to replace placeholders (like `<YOUR_VALUE_HERE>`) with actual values relevant to your setup.

Create a file named `db_install.rsp` and include the following content:

```
[GENERAL]
oracle.install.responseFileVersion=/oracle/install/rspfmt_dbinstall_response_schema_v19.0.0
oracle.install.option=INSTALL_DB_SWONLY
ORACLE_HOSTNAME=<YOUR_HOSTNAME>
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
SELECTED_LANGUAGES=en
ORACLE_HOME=/u01/app/oracle/product/19.0.0/dbhome_1
ORACLE_BASE=/u01/app/oracle

[INSTALL]
nodelist=<YOUR_HOSTNAME>
oracle.install.db.InstallEdition=EE
oracle.install.db.OSDBA_GROUP=dba
oracle.install.db.OSOPER_GROUP=dba
oracle.install.db.OSBACKUPDBA_GROUP=dba
oracle.install.db.OSDGDBA_GROUP=dba
oracle.install.db.OSKMDBA_GROUP=dba
oracle.install.db.OSRACDBA_GROUP=dba
oracle.install.db.rootconfig.executeRootScript=true
```

```
oracle.install.db.config.starterdb.type=GENERAL_PURPOSE
oracle.install.db.config.starterdb.globalDBName=<YOUR_GLOBAL_DBNAME>
oracle.install.db.config.starterdb.SID=<YOUR_SID>
oracle.install.db.ConfigureAsContainerDB=false
oracle.install.db.config.PDBName=<YOUR_PDB_NAME>
oracle.install.db.config.starterdb.characterSet=<YOUR_CHARACTER_SET>
oracle.install.db.config.starterdb.memoryOption=true
oracle.install.db.config.starterdb.memoryLimit=<MEMORY_LIMIT_IN_MB>
oracle.install.db.config.starterdb.installExampleSchemas=false
oracle.install.db.config.starterdb.password.ALL=<YOUR_PASSWORD>
oracle.install.db.config.starterdb.storageType=FS
oracle.install.db.config.starterdb.dataLocation=/u01/oradata
oracle.install.db.config.starterdb.recoveryLocation=<YOUR_RECOVERY_LOCATION>
oracle.install.db.config.starterdb.enableSecuritySettings=true
oracle.install.db.config.starterdb.registerWithDirService=false
oracle.install.db.config.starterdb.listeners=<LISTENER_NAME>
```

[DELPHIX]

```
DELPHIX_DB_CONF=<DELPHIX_CONFIGURATION_OPTION>
```

Replace the placeholders with your specific configuration:

- `<YOUR_HOSTNAME>`: The hostname of your server.
- `<YOUR_GLOBAL_DBNAME>`: The global database name.
- `<YOUR_SID>`: The system identifier for your database.
- `<YOUR_PDB_NAME>`: The pluggable database name, if you're using the multitenant architecture.
- `<YOUR_CHARACTER_SET>`: The character set for your database (e.g., `AL32UTF8`).
- `<MEMORY_LIMIT_IN_MB>`: The memory limit for your database in megabytes.
- `<YOUR_PASSWORD>`: The password for SYS, SYSTEM, and other default administrative accounts.
- `<YOUR_RECOVERY_LOCATION>`: The path for the recovery area.
- `<LISTENER_NAME>`: The name of the Oracle Net listener.
- `<DELPHIX_CONFIGURATION_OPTION>`: Specific to your Delphix configuration, if applicable.

To run the Oracle Database installer silently using your response file, execute the following command from the directory where your Oracle Database installation files are located:

```
./runInstaller -silent -responseFile /path/to/db_install.rsp -ignorePrereqFailure
```

Replace `/path/to/db_install.rsp` with the actual path to your response file.

### Important Notes:

- Ensure that all prerequisites for Oracle Database installation are met before running the installer, including any required packages, kernel parameters, and user and group configurations.
- The `-ignorePrereqFailure` flag is optional and allows the installer to continue even if some prerequisites are not met. It's generally recommended to address all prerequisite checks for a production environment.
- Running the root scripts (`/u01/app/oraInventory/orainstRoot.sh` and `/u01/app/oracle/product/19.0.0/dbhome_1/root.sh`) as the root user is typically required to complete the installation. The installer will prompt you for this if not included in the response file.
- This example does not cover all possible configurations and options. Review the Oracle Database Installation Guide for your specific version and adjust the response file accordingly.

## Database Creation

To create a new Oracle Database that includes a new Container Database (CDB) with a Pluggable Database (PDB), and to ensure that logs, archive logs, data files, and the Fast Recovery Area are on different volumes, you will use a response file tailored for the database creation process using the Database Configuration Assistant (DBCA).

The character set compatible with WebLogic SOA Suite RCU is typically AL32UTF8, as it supports Unicode and is widely used for applications requiring extensive character support.

Given the specifications, the response file will be crafted to meet the following requirements:

- Create a new CDB and PDB
- Allocate 6GB of RAM to the database and enable Automatic Memory Management
- Place logs, archive logs, data files, and the Fast Recovery Area on different volumes
- Use `AL32UTF8` as the character set to ensure compatibility with WebLogic SOA Suite RCU

Below is an example response file template for creating a new Oracle CDB with a PDB. Please adjust paths and names to suit your environment.

```
[GENERAL]
responseFileVersion=/oracle/assistants/dbca/rspfmt_dbca.rsp
GDBNAME = "cdb1.example.com"
SID = "cdb1"
CREATEASCONTAINERDATABASE = true
NUMBEROFPDBS = 1
PDBNAME = "pdb1"
PDBADMINPASSWORD = "<Your_PDB_Admin_Password>"
TEMPLATENAME = "General_Purpose.dbc"
```

```
CHARACTERSET = "AL32UTF8"
NATIONALCHARACTERSET= "AL16UTF16"
DATABASETYPE = "MULTIPURPOSE"
```

```
[createDatabase]
gdbName = cdb1
sid = cdb1
createAsContainerDatabase=true
numberOfPDBs=1
pdbName=pdb1
templateName=General_Purpose.dbc
characterSet=AL32UTF8
nationalCharacterSet=AL16UTF16
databaseType=MULTIPURPOSE
automaticMemoryManagement=true
totalMemory=6000
storageType=FS
datafileDestination=/u02/oradata
recoveryAreaDestination=/u03/fast_recovery_area
redoLogFileSize=300
emConfiguration=NONE
listeners=LISTENER
```

```
[createContainerDatabase]
gdbName = cdb1
templateName = General_Purpose.dbc
sid = cdb1
createAsContainerDatabase = true
numberOfPDBs = 1
pdbName = pdb1
createPDBTemplate = "PDB Admin Managed Template"
pdbAdminUserName = admin
pdbAdminPassword = <Your_PDB_Admin_Password>
datafileDestination = /u02/oradata/cdb1/
recoveryAreaDestination = /u03/fast_recovery_area/cdb1/
storageType = FS
```

```
[CONFIGUREDATABASE]
LOGFILEDEST_1 = "/u04/logs/"
LOGFILEDEST_2 = "/u05/archive_logs/"
```

```
LOGFILEDEST_3 = "/u02/oradata/"
```

Replace `<Your_PDB_Admin_Password>` with a secure password for your PDB admin user.

## Notes:

- **Paths:** `/u02/oradata`, `/u03/fast_recovery_area`, `/u04/logs`, and `/u05/archive_logs` should be replaced with the actual mount points or directories you've set up on your system for data files, the Fast Recovery Area, logs, and archive logs, respectively.
- **Total Memory:** The `totalMemory` parameter is set to 6000, representing approximately 6GB of RAM dedicated to the Oracle instance. Oracle will manage this memory automatically because `automaticMemoryManagement` is set to true.
- **Character Sets:** `CHARACTERSET` is set to `AL32UTF8`, and `NATIONALCHARACTERSET` is set to `AL16UTF16`, which are suitable for globalized applications and compatibility with WebLogic SOA Suite RCU.
- **Listeners:** This template assumes a listener named `LISTENER` is already configured on the default port (1521). Adjust as necessary for your environment.

## Execution

To create the database using the response file, you would typically use the `dbca` command-line tool, specifying the response file with the `-silent` and `-responseFile` options:

```
dbca -silent -createDatabase -responseFile /path/to/your_response_file.rsp
```

Make sure to replace `/path/to/your_response_file.rsp` with the actual path to your response file.

**Important:** This response file template is provided as a starting point. You may need to adjust values and paths according to your specific Oracle Database version, system architecture, and requirements. Always review the Oracle Database documentation for the version you're installing to ensure compatibility and correctness.

# Oracle XE and APEX on CentOS 7

## Downloading the software

The first thing to do is download the software from Oracle Technology Network:

- [Database Downloads](#) - you will need *the package for Linux x64* and the *preinstall RPM package*.
- [Developer Tools/Oracle REST Data Services/Downloads](#)
- [Developer Tools/Application Express/Downloads](#)

After the files have been downloaded, transfer them to the server.

## Installation of RDBMS

After you checked them, to install the RDBMS, you need to install the *preinstall RPM package* first and then install the database software as following:

```
wget https://yum.oracle.com/repo/OracleLinux/OL7/latest/x86_64/getPackage/oracle-database-preinstall-18c-1.0-1.el7.x86_64.rpm
yum install oracle-database-preinstall-18c* -y
yum install oracle-database-xe-18c* -y
yum install httpd tomcat -y
```

The user `oracle` and the group `oinstall` are created during the package installation. A default user environment is created during the set up process. You can set a password for this user by invoking `passwd oracle` command. This user is the owner of the `/opt/oracle` directory where the Oracle Database is located and this must stay unchanged.

```
chown oracle:oinstall /opt/oracle
```

When the packages are installed and the user is set up, you need to run the initial database configuration script and answer all of the questions.



```
/etc/init.d/oracle-xe-18c configure
```

After answering the questions it is going to take several minutes to initialize the database.

## Setting up environment

Set up Oracle Database environment variables .

```
echo 'ORACLE_SID=XE' >> /etc/profile.d/oraenv.sh
echo 'ORAENV_ASK=NO' >> /etc/profile.d/oraenv.sh
echo '. /opt/oracle/product/18c/dbhomeXE/bin/oraenv -s' >> /etc/profile.d/oraenv.sh
. /etc/profile.d/oraenv.sh
```

Enable Oracle Database XE service for automatic startup:

```
systemctl enable oracle-xe-18c
```

## Connecting to database

And we are ready to log into the database.

```
sqlplus /nolog
```

Check if everything is good.

```
-- connect to the database
sqlplus /nolog

-- change role
CONNECT SYS as SYSDBA

-- basic query to check everything came up right
select * from dual;

-- exit the database
exit
```

To make it easier to connect to the pluggable database, edit thetnsnames.ora file and add there a new connection descriptor



```
        privilege => 'resolve');
dbms_network_acl_admin.assign_acl(acl => 'all-network-PUBLIC.xml',
        host => '*');

END;
/
sho err
COMMIT;
/
```

From the APEX directory connect to the pluggable database as `sysdba` and run the installation scripts.

```
cd /opt/oracle/apex
-- connect to the database
sqlplus /nolog

-- change role
CONN sys@pdb1 AS SYSDBA
```

```
-- run the script to install a full development environment
@apexins.sql SYSAUX SYSAUX TEMP /i/

-- create an instance administrator user and set their password
@apxchpwd.sql

-- unlock APEX public user
ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK;
ALTER USER APEX_PUBLIC_USER IDENTIFIED BY "S0m3aw3s0m3Pw!";

-- configure REST Data Services
@apex_rest_config.sql

-- run the ACL setup
@apex_acl.sql

-- disconnect from the database
exit
```

Copy APEX static files to the web server directory. They will be used to serve static images from the proxy.

```
mkdir -p /var/www/apex/images
cp -a /opt/oracle/apex/images/. /var/www/apex/images
```

The Application Express installation is complete.

# Installation of ORDS

The Oracle Rest Data Services (ORDS) installation consists of unzipping the downloaded archive, running the configuration command, and then deploying the `ords.war` file into the Tomcat webapps folder.

```
cd /root
mkdir -p /opt/oracle/ords
unzip ords-19.*.zip -d /opt/oracle/ords
```

Run the ORDS configuration command with the advanced mode to run the interactive installation process.

```
cd /opt/oracle/ords
java -jar ords.war install advanced
```

When prompted for *ORDS* configuration directory, enter `config`.  
Then provide the connection info to your pluggable database `XEPDB1`

Follow the on screen instructions.

After the configuration is completed, the values are saved in `opt/oracle/ords/config/ords/defaults.xml` file. It can be modified there. See more at Oracle Docs.

The `tomcat` user (created as part of *Tomcat* install) must have read-write access to the *ORDS* configuration folder:

```
chown -R tomcat:tomcat /opt/oracle/ords/config
```

Deploy *ORDS* to Tomcat application server. Copy the `ords.war` into the *Tomcat* `webapps` directory for this

```
cp -a /opt/oracle/ords/ords.war /usr/share/tomcat/webapps/
```

Done with ORDS and Tomcat, on to Apache.

# Configuration of Apache httpd to map ORDS

The last step is to configure Apache to map HTTP-requests to ORDS and therefore APEX engine.

For this, add a custom `httpd` configuration file. By default, every `.conf` file placed in the `etc/httpd/conf.d/` directory is read by `httpd` as an additional configuration file to the main `/etc/httpd/conf/httpd.conf` config file.

Create the `apex.conf` file in the `etc/httpd/conf.d/` directory with the contents as below:

```
# forward ORDS tomcat
<VirtualHost *:80>
    # uncomment the lines below if you plan to serve different domains
    # on this web server, don't forget to change the domain name
    # ServerName yourdomain.tld
    # ServerAlias www.yourdomain.tld

    # alias for APEX static files
    Alias "/i" "/var/www/apex/images/"

    # uncomment the line below if you want
    # to redirect traffic to ORDS from root path
    # RedirectMatch permanent "^/$" "/ords"

    # proxy ORDS requests to tomcat
    ProxyRequests off
    <Location "/ords">
        ProxyPass "ajp://localhost:8009/ords"
        ProxyPassReverse "ajp://localhost:8009/ords"
    </Location>
</VirtualHost>
```

Tell SELinux (Yes, that should be running) to allow Apache to communicate to tomcat.

```
setsebool httpd_can_network_connect on
```

Now you are ready to save the configuration file and restart the services.

```
systemctl restart httpd  
systemctl restart tomcat
```

Open a few firewall ports. Yes, the the firewall should also be on.

```
firewall-cmd --permanent --add-service={http,https}  
firewall-cmd --reload
```

And finally, access APEX from your web browser using a link like `http://yourdomain.tld/ords` (or `http://yourdomain.tld` in case you switched on force redirection), where `yourdomain.tld` is the domain name or the IP-address of your server.