GLOC

Great Lakes Oracle Conference

Java Best Practices for Developing and Deploying against Database in the Cloud

MAY 16 & 17, 2018

CLEVELAND PUBLIC AUDITORIUM, CLEVELAND, OHIO

WWW.NEOOUG.ORG/GLOC

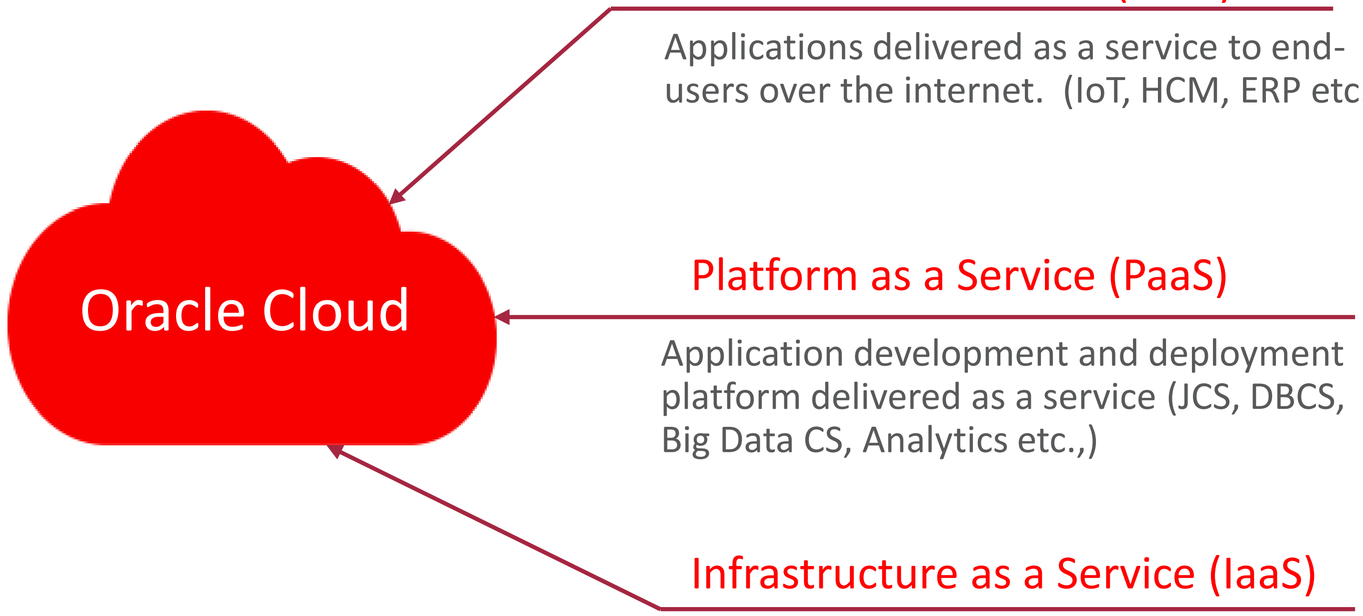
Safe Harbor Statement

Program Agenda

Program Agenda with Highlight

1. What are we talking about?
2. Security Settings
3. Demos
4. Java Best Practices
5. Questions

Oracle Cloud Layers



Software as a Service (SaaS)

Applications delivered as a service to end-users over the internet. (IoT, HCM, ERP etc.,)

Platform as a Service (PaaS)

Application development and deployment platform delivered as a service (JCS, DBCS, Big Data CS, Analytics etc.,)

Infrastructure as a Service (IaaS)

Server, Storage, and Network hardware and associated software delivered as a service

Database Cloud Services for Enterprise

100% compatible from entry-level to the largest mission critical dataloads

	Express CS	Database CS	Oracle Cloud Infra	Exadata CS
Database Development	✓	✓	✓	✓
SMB & Departmental Applications	✓	✓	✓	✓
Enterprise Applications		✓	✓	✓
Highest availability, scalability and performance				✓
Cloud at Customer Deployment		**Cloud at Customer**		**Exadata Cloud at Customer**

What are we talking about?

- Plain Java standalone apps
- Java App Containers
  - Apache Tomcat
  - Oracle Web Logic Server
  - IBM Websphere and JBOSS
- Java tools or IDEs
  - SQLDeveloper, Squirrel SQL
  - Intellij, Jdeveloper, Eclipse, Netbeans
- All these Java apps must be able to connect to a Database Service on Cloud

What are the things to consider?

JDBC drivers must meet the cloud specific requirements

For example:

- Support for TLS v1.2 with unlimited cipher suites
- Protocol specific encryption and checksumming
- Support various keystore formats (KSS, JKS, Wallets)
- Support strong authentication
  - Based on certificates
  - Kerberos authentication

How about these?

- Proper error messages and traces to debug connectivity issues
- Support keepalive mechanisms
- HTTP proxy and websockets

Ideally

- Reconnect on failure and replay in-flight work
- How about asynchronous APIs?
  - The API is available for download from OpenJDK at
  - http://www.oracle.com/goto/java-async-db
  - Send feedback to jdbc-spec-discuss@openjdk.java.net

Program Agenda with Highlight

Java Connectivity to DB Services on Cloud

- **Oracle Exadata Express Service (EECS)**
  - TCPS connection with Oracle wallets or JKS
  - Need **client_credentials.zip** from service console

- **Oracle Database Cloud Service (DBCS)**
  - TCP connection
  - 1521 needs to be unblocked

- **Database on Oracle Cloud Infrastructure (OCI)**
  - TCP connection
  - Add 1521 to security list

- **Oracle Database Exadata Cloud Machine (ExaCM)**
  - TCP connection
  - Everything is pre-setup

Example with Oracle Cloud Database Service

Database Environment	Default Connectivity
Database as a Service (DBaaS)	TCP/IP with network encryption (Port 1521) To allow direct connection, open port 1521 for specific trusted hosts
Exadata Express Cloud Service (EECS) Fully managed	TCPS (Port 1522) TLS v1.2 and strong security algorithms is mandatory Two-stages authentication: Must have client wallet in addition to database credentials

ORACLE

Security Settings

Mandatory: using latest JDK with JCE

- JDK version is important
  - Security bugs in some older releases
  - Always use the latest JDK upgrade
- JCE Unlimited Strength Jurisdiction Policy files
  - JDK 9 has JCE by default
  - Need to be installed in the Java runtime for JDK 7 and JDK 8.
  - TLS_RSA_WITH_AES_256_GCM_SHA384 and TLS_RSA_WITH_AES_256_CBC_SHA256 cipher suites use AES with 256 bit keys and hence require Unlimited JCE policy files

Security Settings using Oracle Wallets

- Additional jars are needed
  - **oraclepki.jar , osdt_core.jar, and osdt_cert.jar**
  - Available on Oracle Maven repository (maven.oracle.com)
- Make sure to have wallets at an accessible location
  - **cwallet.sso (auto-login format)** or **ewallet.p12 (PKCS12 format)**
- Provide the location of the wallet
  - **oracle.net.wallet_location**=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/Users/test/wallets/)))
- Enforce mutual authentication
  oracle.net.ssl_server_dn_match=true

Security Settings using Java Key Store (JKS)

- Configure trustStore and keyStore
  - Use **javax.net.ssl.trustStore** and **javax.net.ssl.keyStore** system properties or connection properties
- Set the password for JKS
  - Use **javax.net.ssl.keyStorePassword** and **javax.net.ssl.trustStorePassword**
- Enable the server DN match
  - **oracle.net.ssl_server_dn_match=true**

Connecting to the Cloud is easier than ever

NEW IN 18c

- New **ojdbc.properties** for specifying connection properties
  - Introduced specifically to ease the cloud connectivity
  - If TNS_ADMIN system property is defined, then it becomes the default location for ojdbc.properties
- Using Oracle Wallets with JDBC thin is made simpler
  - Auto-load of **oraclepki.jar, osdt_cert.jar** and **osdt_core.jar**
  - OraclePKIProvider is automatically registered
- TNS_ADMIN can be set in the URL
  - jdbc:oracle:thin:@alias**?TNS_ADMIN=/home/oracle/network/admin/**

Program Agenda with Highlight

1. What are we talking about?
2. Security Settings
3. Demos
4. Java Best Practices
5. Questions

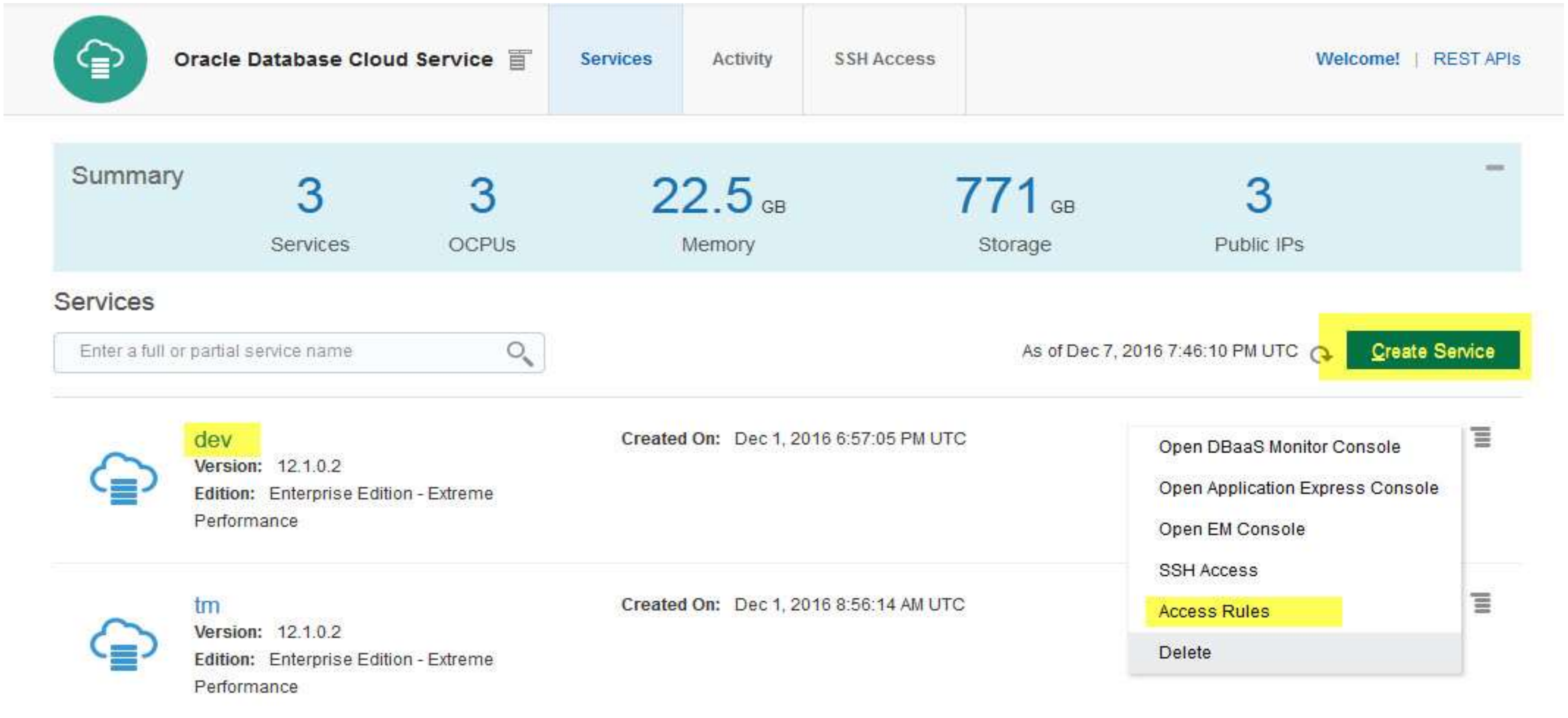
ORACLE

DBCS Connectivity – Overview

- **TCP connections** allowed
  - Port 1521 needs to be unblocked before usage
- Full control over the database.
  - HR schema is available, but needs to be unlocke
  - Create more users or schemas or tables by connecting to the compute node
- SSH access to the compute node

Java connectivity to DBCS

Create the service and unblock port 1521

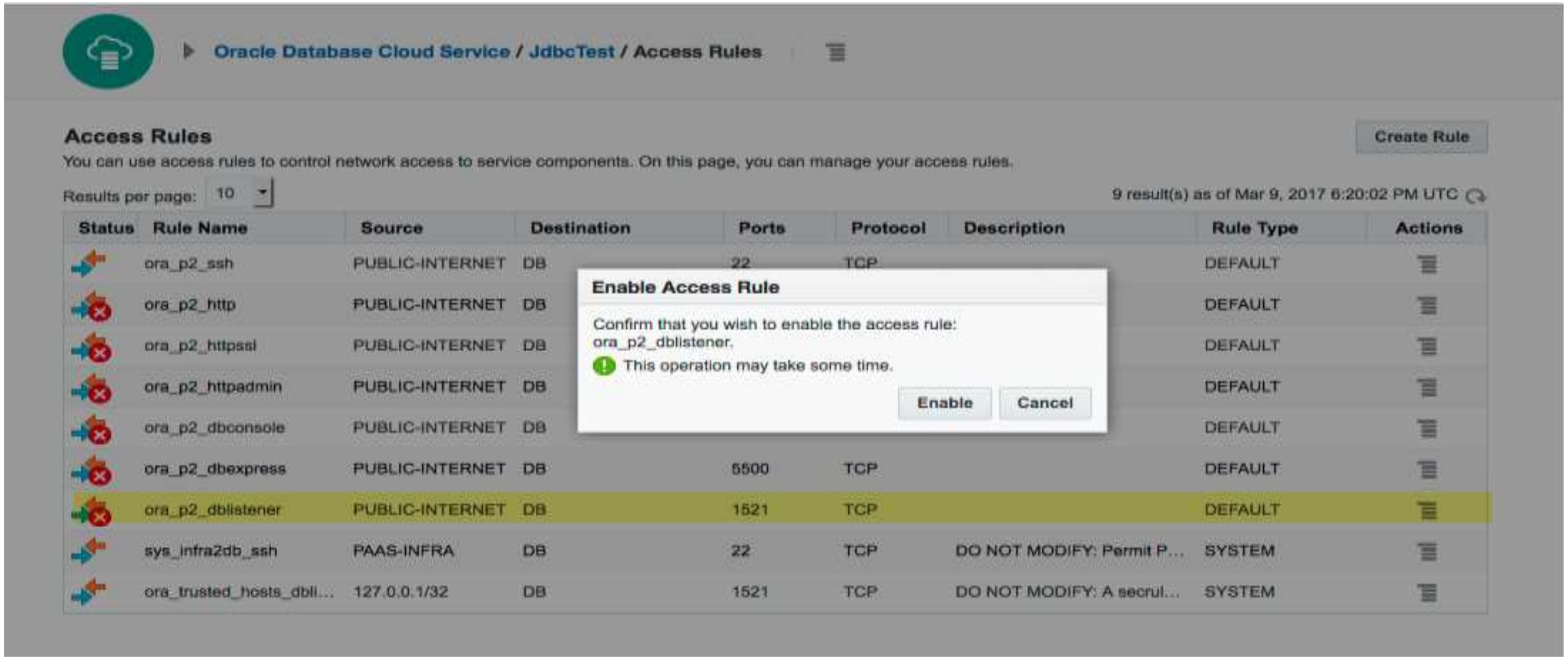


dev
Version: 12.1.0.2
Edition: Enterprise Edition - Extreme Performance
Created On: Dec 1, 2016 6:57:05 PM UTC

tm
Version: 12.1.0.2
Edition: Enterprise Edition - Extreme Performance
Created On: Dec 1, 2016 8:56:14 AM UTC

Java Connectivity to DBCS

Unblock the port 1521



Oracle Database Cloud Service / JdbcTest / Access Rules

Access Rules

You can use access rules to control network access to service components. On this page, you can manage your access rules.

Results per page: 10

9 result(s) as of Mar 9, 2017 6:20:02 PM UTC

Status	Rule Name	Source	Destination	Ports	Protocol	Description	Rule Type	Actions
	ora_p2_ssh	PUBLIC-INTERNET	DB	22	TCP		DEFAULT	
	ora_p2_http	PUBLIC-INTERNET	DB				DEFAULT	
	ora_p2_httpssl	PUBLIC-INTERNET	DB				DEFAULT	
	ora_p2_httpadmin	PUBLIC-INTERNET	DB				DEFAULT	
	ora_p2_dbconsole	PUBLIC-INTERNET	DB				DEFAULT	
	ora_p2_dbexpress	PUBLIC-INTERNET	DB	5500	TCP		DEFAULT	
	ora_p2_dblistener	PUBLIC-INTERNET	DB	1521	TCP		DEFAULT	
	sys_infra2db_ssh	PAAS-INFRA	DB	22	TCP	DO NOT MODIFY: Permit P...	SYSTEM	
	ora_trusted_hosts_dbli...	127.0.0.1/32	DB	1521	TCP	DO NOT MODIFY: A secrul...	SYSTEM	

Enable Access Rule

Confirm that you wish to enable the access rule: ora_p2_dblistener.

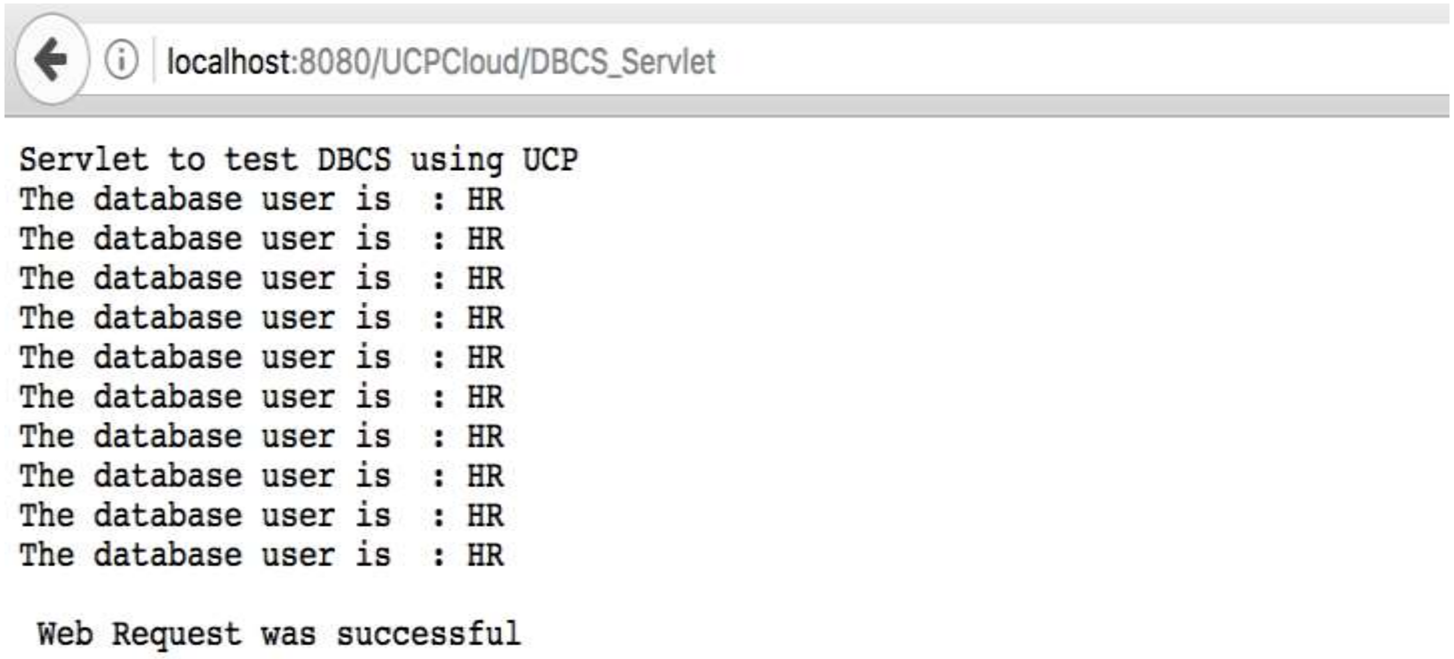
This operation may take some time.

Java Connectivity to DBCS using Tomcat

Sample context.xml

```
<Context>
 <Resource name="tomcat/UCPPool_dbcs" auth="Container"
factory="oracle.ucp.jdbc.PoolDataSourceImpl"
type="oracle.ucp.jdbc.PoolDataSource"
description="UCP Pool in Tomcat"
connectionFactoryClassName="oracle.jdbc.pool.OracleDataSource"
minPoolSize="5"   maxPoolSize="50"   initialPoolSize="15"
autoCommit="true"   user="hr"   password="hr”
url="jdbc:oracle:thin:@140.86.13.179:1521/PDB1.dbdevcs14.oracl
ecloud.internal" />
```

Screenshot of connecting to DBCS



```
Servlet to test DBCS using UCP
The database user is  : HR
The database user is  : HR
The database user is  : HR
The database user is  : HR
The database user is  : HR
The database user is  : HR
The database user is  : HR
The database user is  : HR
The database user is  : HR
The database user is  : HR

 Web Request was successful
```

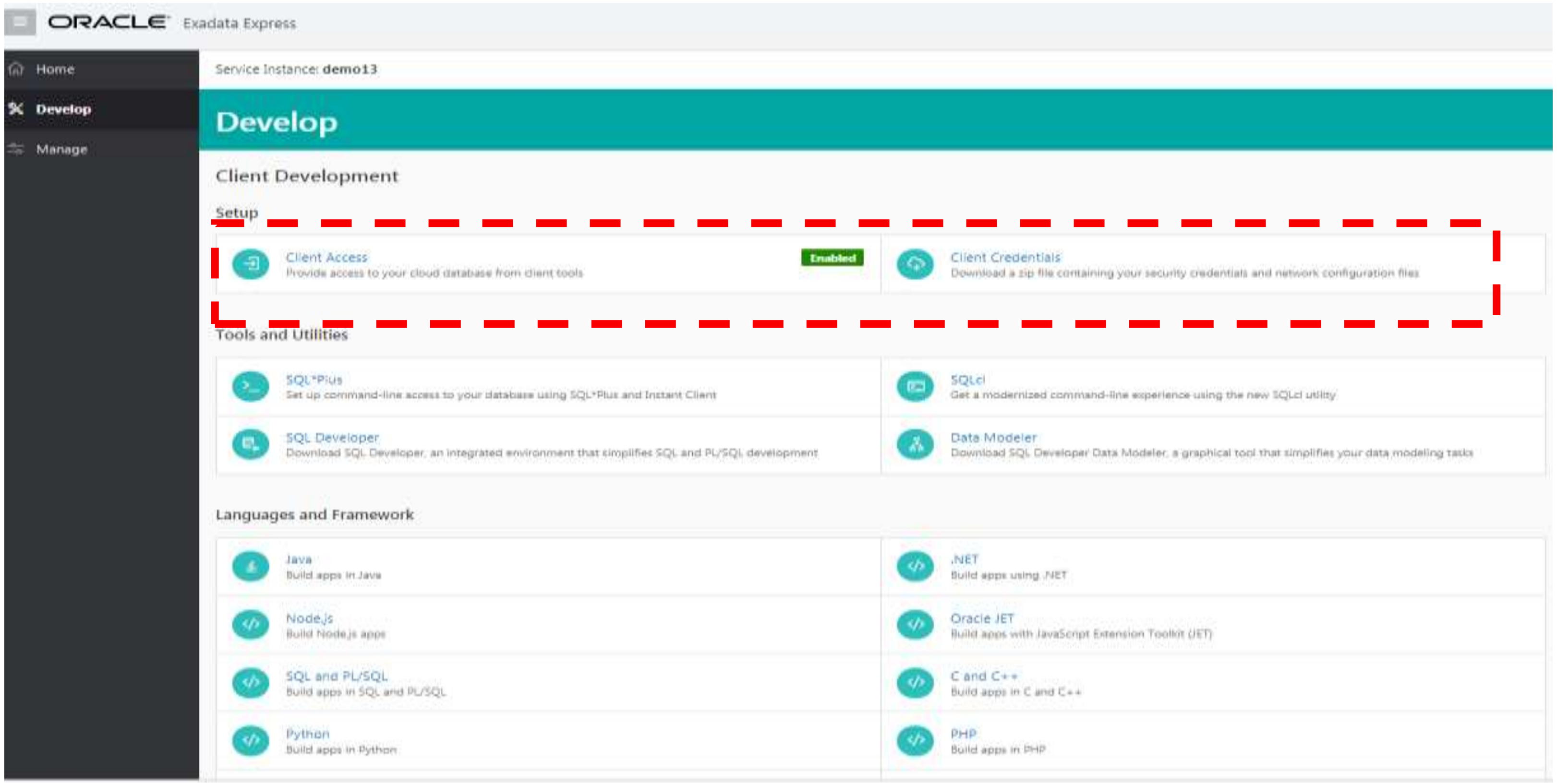
EECS Connectivity – Overview

A Fully Managed experience for hands-free cloud database operation

- TCPS connections required
- Mandates SSL connection using **TLSv1.2**
  - Java Key Store Files or Oracle Wallets
- PDB_ADMIN is the user created by default
  - Create your own user
- Requires Java Cryptography Extension (JCE) in the JDK/JRE.

Exadata Express Cloud Service Connectivity

Download client_credentials.zip



Develop

Client Development

Setup

- Client Access — Enabled — Provide access to your cloud database from client tools
- Client Credentials — Download a zip file containing your security credentials and network configuration files

Tools and Utilities

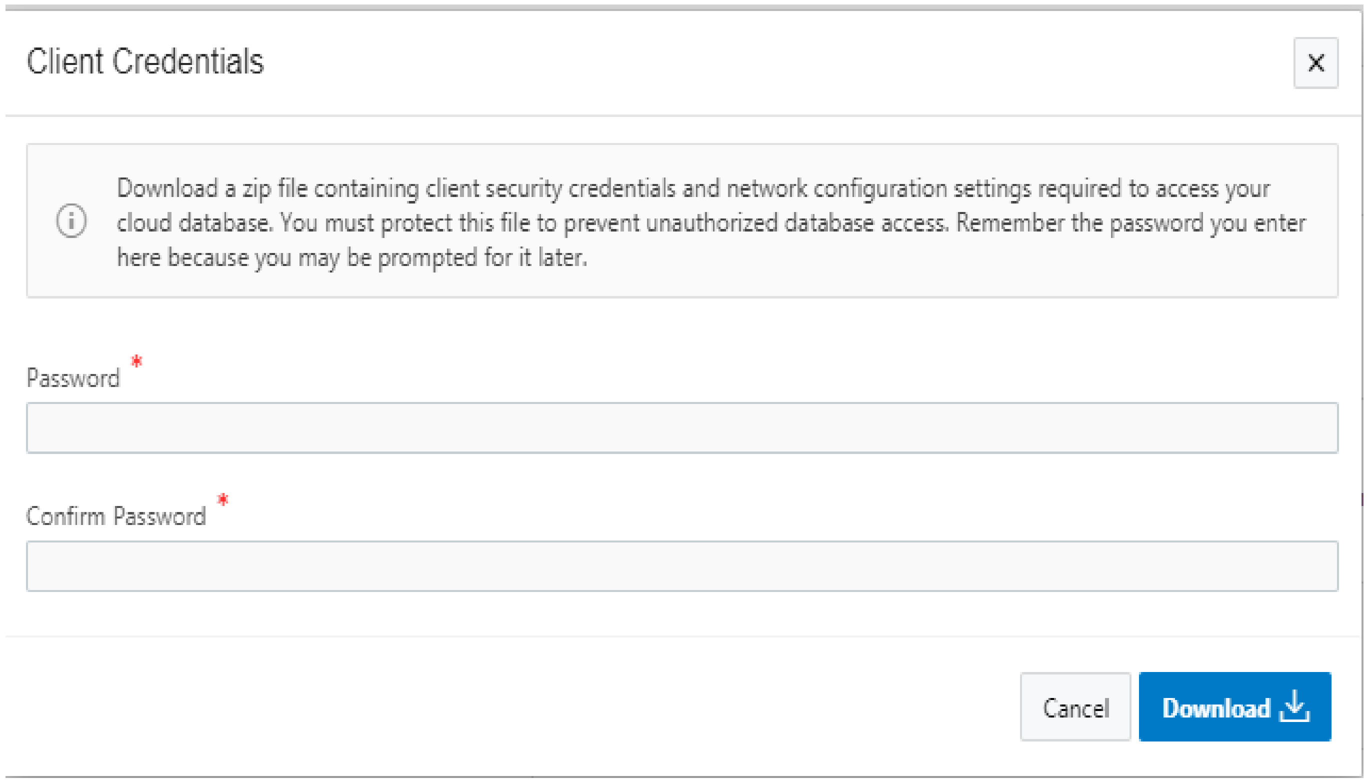
- SQL*Plus — Set up command-line access to your database using SQL*Plus and Instant Client
- SQLcl — Get a modernized command-line experience using the new SQLcl utility
- SQL Developer — Download SQL Developer, an integrated environment that simplifies SQL and PL/SQL development
- Data Modeler — Download SQL Developer Data Modeler, a graphical tool that simplifies your data modeling tasks

Languages and Framework

- Java — Build apps in Java
- .NET — Build apps using .NET
- Node.js — Build Node.js apps
- Oracle JET — Build apps with JavaScript Extension Toolkit (JET)
- SQL and PL/SQL — Build apps in SQL and PL/SQL
- C and C++ — Build apps in C and C++
- Python — Build apps in Python
- PHP — Build apps in PHP

EEC S Connectivity

Choose wallet or keystore password


Client Credentials

Download a zip file containing client security credentials and network configuration settings required to access your cloud database. You must protect this file to prevent unauthorized database access. Remember the password you enter here because you may be prompted for it later.

Password *

Confirm Password *

Exadata Express Connectivity

client_credentials.zip contents

File name	Description
tnsnames.ora and sqlnet.ora	Network configuration files storing connect descriptors and SQL*Net client side configuration
cwallet.sso and ewallet.p12	Auto-open SSO wallet and PKCS 12 file. PKCS 12 file is protected by the wallet password provided in the UI.
truststore.jks and keystore.jks	JKS Truststore and Keystore. Protected by the wallet password provided in the UI.

EECS Connectivity

Pre-requisites

- For Thin JDBC
  - Unzip the **client_credentials.zip** file to any location
  - Update JDK path to use the latest JDK8/JDK7 with the required JCE policy files
  - Pass truststore or wallet related parameters as connection/system properties
  - Connect using the connection string **“jdbc:oracle:thin:@dbaccess”** with dbaccess being the TNS alias.

Detailed steps are documented in Exadata Express Service Console links

Exadata Express Cloud Service Connectivity

Sample script to run

```
java -Doracle.net.tns_admin=/home/myuser/cloud \
    -Doracle.net.ssl_server_dn_match=true \
    -Djavax.net.ssl.trustStore=/home/myuser/cloud/truststore.jks \
    -Djavax.net.ssl.trustStorePassword=welcome1 \
    -Djavax.net.ssl.keyStore=/home/myuser/cloud/keystore.jks \
    -Djavax.net.ssl.keyStorePassword=welcome1 \
EECSDataSourceSample
```

Exadata Express Cloud Service Connectivity

NEW IN 18c

Uses Oracle Wallet

- Required security settings are pushed to **ojdbc.properties** file
- Pass the TNS_ADMIN in the connection URL
  Example:
  jdbc:oracle:thin:@alias**?TNS_ADMIN=/home/oracle/network/admin/**
- java –classpath ./lib/ojdbc8_181.jar:.:./lib/oraclepki.jar:./lib/osdt_cert.jar:./lib/osdt_core.jar EECSDataSourceSample_Wallet

Program Agenda with Highlight

ORACLE

Java Best Practices

Connecting to Database services on Cloud

- Best Practices for Performance
- Best Practices for Security
- Best Practices for High Availability
- Troubleshooting tips

Best Practices for Performance

Reduce round trips, optimize sessions and data transfer

- Use Connection Pooling (Example: UCP)
  - Optimize Min Pool Size, Max Pool Size and timeouts
- Bind variables
  - Prevents re-parsing of frequently executed statements
  - Re-execute the same PreparedStatement with different binds
- Array operations instead of single row operations
  - DML Batching and Row Prefetch
  - **preparedStatement.addBatch() and preparedStatement.sendBatch()**

Best Practices for Performance

Reduce round trips, optimize sessions and data transfer

- Prefetching
  - Prefetch a number of rows (configurable) **preparedStatement.setFetchSize(20)**
- Statement Caching
  - Caches most recently used statements
  - `oracleDataSource`.**`setImplicitCachingEnabled(true)`** **and** `connection`.**`setStatementCacheSize(10)`**
- Client Query Result Cache
  - Caches SQL query results on client tier
  - Oracle transparently maintains cache consistency with server side changes

Best Practices for Performance

Reduce round trips, optimize sessions and data transfer

- Co-locate application servers and database servers (if possible) in order to reduce latency
  - Run ping or traceroute to look at latency
- Tune the Session Data Unit(SDU) for large LOBs, XMLs, large resultSets
  - Max: **2MB (12c)**, 64K (11.2), 32K (pre-11.2)
    - Set on both server and client side (sqlnet.ora(DEFAULT_SDU_SIZE), tnsnames.ora or URL)
    - jdbc:oracle:thin:@(DESCRIPTION=**(SDU=11280)** (ADDRESS=(PROTOCOL=tcp)(HOST=myhost-vip)(PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=myorcldbservicename)))
- Sharded database for scalability

Best Practices for Security

- For corporate environment, use VPN
- If you want to enable direct connection
  - Enable access to Database listeners from only specific set of trusted IP addresses
- Set up Logon Storm Handler to limit the connection rate
  - RATE_LIMIT parameter for Listener
- Protect the wallet or keystore
  - Ensure that the files are protected through file system permissions, backed up securely, and only read access is granted to the users running applications at run-time

Best Practices for High Availability

- Leverage most advanced HA features by using latest DB client
  - Timeout and retry in connect string
  - Application Continuity
    - Replay Driver
    - At most once commit
    - Inflight transactions are transparently replayed in case of failure
  - Fast Application Notification
    - More reliable and predictable than the ugly TCP timeout
    - In band notifications are preferred

Oracle Best Practices for High Availability

Gracefully handle service temporary unavailability

Enable TCP Keep Alive

TCP/IP level timeout

Retry while service is unavailable

```
(DESCRIPTION_LIST =
  (DESCRIPTION=
   (ENABLE=BROKEN)
    (TRANSPORT_CONNECT_TIMEOUT=10)
    (RETRY_COUNT=10)(RETRY_DELAY=5)
      (ADDRESS_LIST=(ADDRESS = .   .   .)(ADDRESS= .  .  .))
    (CONNECT_DATA=(SERVICE_NAME=hr_svc)))
   (DESCRIPTION=
    (RETRY_COUNT=10)(RETRY_DELAY=5)
    (ADDRESS_LIST=(ADDRESS = .   .   .)(ADDRESS=.  .  .))
    (CONNECT_DATA=(SERVICE_NAME=hr_svc2))))
```

ORACLE

Trouble shooting Tips

- Common cause
  - Firewalls blocking connections
- Troubleshooting
  - Run traceroute, e.g.
    - traceroute -T -p 1521 <IP of DBaaS host> (for DBaaS)
      (You can find DBaaS host Public IP from DBaaS Service Console)
    - traceroute -T -p 1522 <public host name for your Exadata Express Cloud Service>
      (You can find the public host name from the tnsnames.ora file, which is included in the zip file downloaded from Service Console. Example: dbaccess.us2.oraclecloudapps.com)
  - Identify where it is failing and take appropriate actions

ORACLE

Program Agenda with Highlight

Questions?

Integrated Cloud

Applications & Platform Services