



ORACLE
OPEN
WORLD

Modern Business in the Cloud

Find, Fix and Validate using Oracle
Enterprise Manager 13c

ORACLE®

Table of Contents

Database Performance Management On-Premises & OCI User Managed DB Systems.....	3
A. Performance Hub.....	7
B. Real-Time Database Operations Monitoring.....	16
C. Tuning a SQL in a PDB.....	19
Real Application Testing	24
D. SQL Performance Analyzer Optimizer Statistics.....	25
E. Database Replay (Optional, time permitting).....	39

Database Performance Management On-Premises & OCI User Managed DB Systems

Objective:

The objective of this lab to provide exercises designed to showcase the new Database Performance Management capabilities in Oracle Enterprise Manager Cloud Control 13c that supports On-premises and Oracle Cloud databases (VM/BM/ExaCS).

Functional Coverage:

In this lab you will go through features in the following functional areas:

- **Performance Hub**
- **Real-time database operation monitoring**
- **Tuning a SQL in a PDB**
- **SQL Performance Analyzer Optimizer Statistics**
SQL Performance Analyzer Gather Optimizer Statistics Validation
- **Consolidated Database Replay**

Replay multiple workloads concurrently against Pluggable Databases in the Container Database

- **Appendix A**

Exercise overview and Business Case

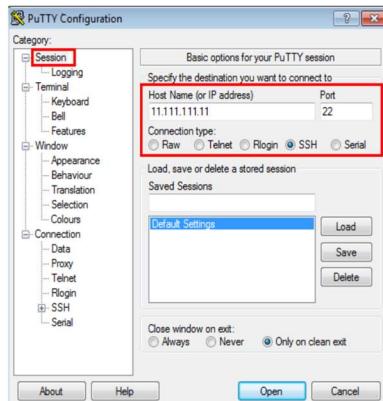
Lab environment setup

What Do You Need?

- Windows or Mac
- PuTTY as an SSH client if using Windows
- Your service instance's public IP address.

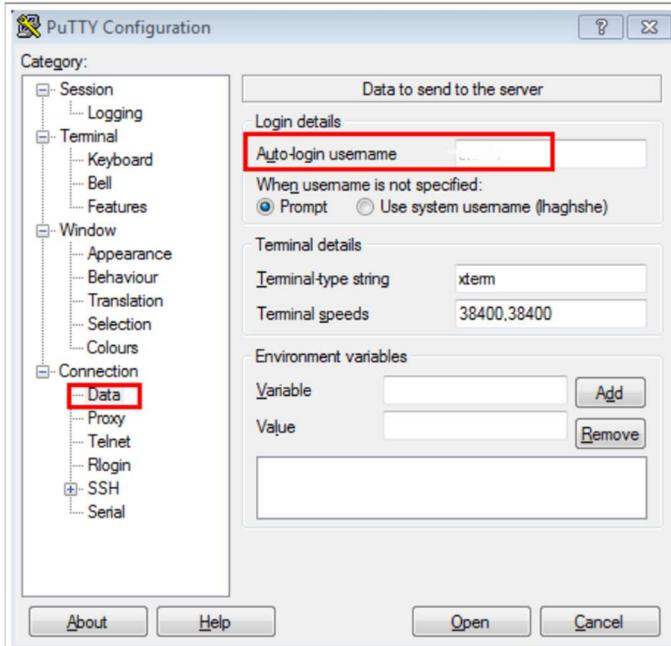
On Windows, you can use PuTTY as an SSH client. PuTTY allows Windows users to connect to remote systems over the internet using SSH and Telnet. SSH is supported in PuTTY, provides for a secure shell, and encrypting information before it's transferred.

1. Download and install PuTTY.
 - a. Go to <http://www.putty.org/> and click the **You can download PuTTY here** link.
2. Run the PuTTY program.
 - a. On your computer, go to **All Programs > PuTTY > PuTTY**
3. Select or enter the following information:
 - **Category:** Session
 - **IP address:** Your service instance's public IP address
 - **Port:** 22
 - **Connection type:** SSH



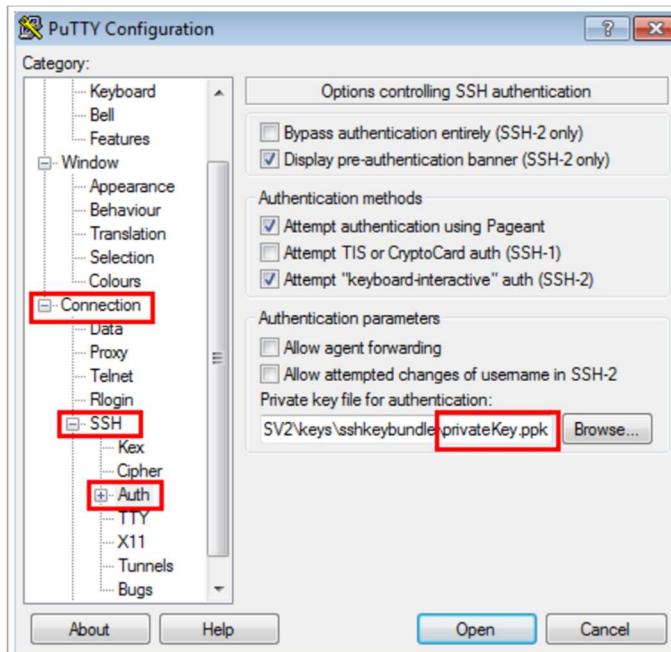
Configuring Your Automatic Login

1. In the **Category** section, click **Connection** and then select **Data**.
2. As your **Auto-login username**, enter opc.



Adding Your Private Key

1. In the **Category** section, click **Auth**.
2. Click **Browse** and find the private key file that matches your VM's public key. This private key should have a **.ppk** extension for PuTTy to work.



To save all your settings:

- In the **Category** section, click **Session**.
- In the **Saved Sessions** section, name your session **EM13C-ABC** and click **Save**.

Connecting to Your VM from a Unix-style System (Mac)

1. Use the following command to set the file permissions so that only you can read the file:
 - \$ chmod 400 <private_key>
 - <*private_key*> is the full path and name of the file that contains the private key associated with the instance you want to access.
2. Use the following SSH command to access the instance
 - ssh -i <private_key> opc@<public-ip-address>
 - <*public-ip-address*> is your instance IP address that you retrieved from the Console.

Running your Workload

1. Change to oracle user via sudo
 - \$ sudo su - oracle
2. Change directory to scripts
 - \$ cd scripts
 - Set the environment for the database by running
 - \$ source SALESENV
3. Execute the script 1-db_lab_start.sh as shown below

```
[oracle@em12 scripts]$ pwd  
/home/oracle/scripts  
[oracle@em12 scripts]$ ./1-db_lab_start.sh
```

A. Performance Hub

Estimated Time to Complete Use Case: **15 minutes**

Business Case

Oracle Enterprise Manager 13c includes the Jet based Performance Hub, a completely new unified interface for performance monitoring. It is the single pane of glass view of database performance with access to ASH Analytics, Real-Time SQL Monitoring and SQL Tuning under the same hood. A flexible time picker allows the administrator to seamlessly switch between Real-Time and Historical views of database performance.

In this lab we will use an Oracle Database 18c Container database with few pluggable databases. You will be logging in as a CDBA.

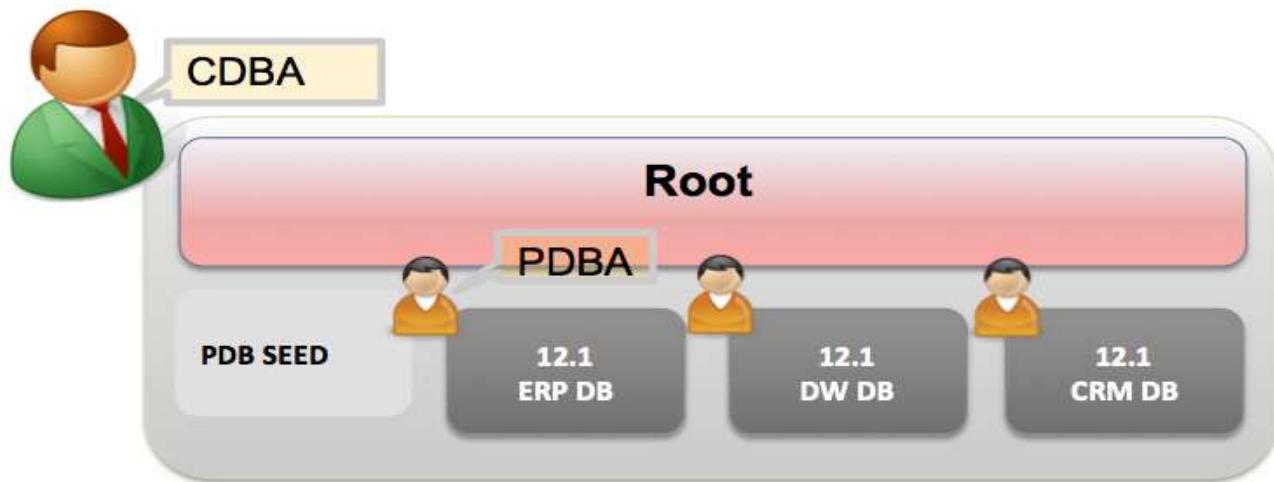


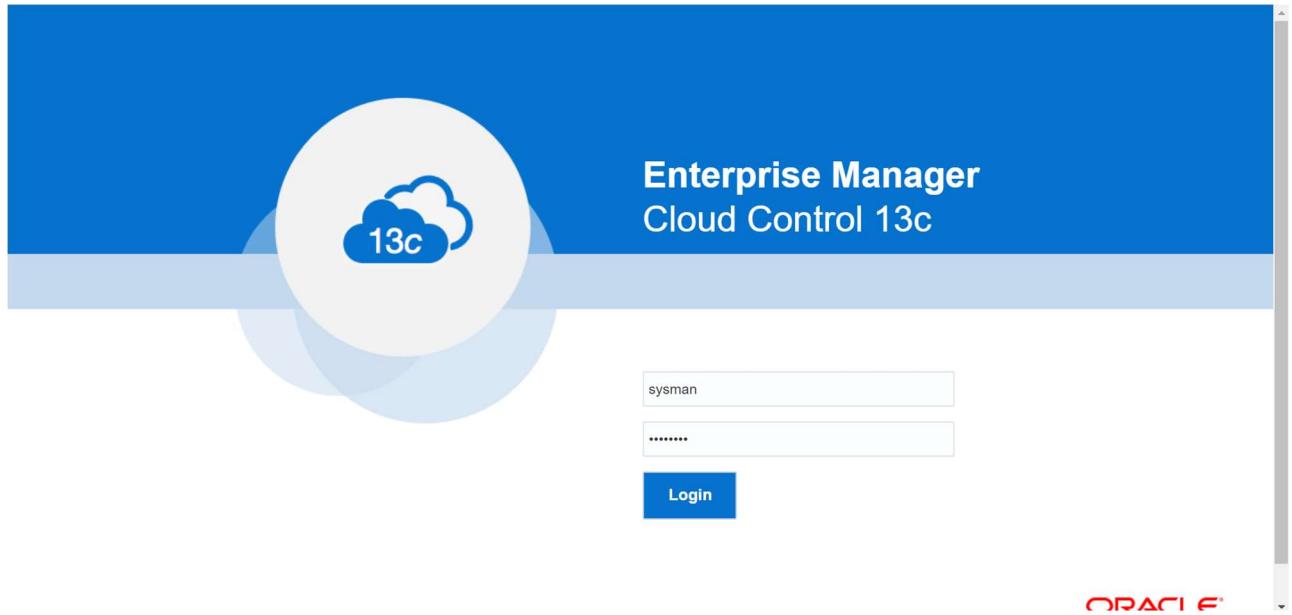
Figure 1: A Container database with PDBs

Understanding the Performance Hub

1.1 Log into Enterprise Manager Please type

<https://<<your assigned i/p address>>: 7803/em/login>

Use the credentials [sysman/welcome1] to log into EM



1.2 Click on the Targets-> Databases. You will be directed to the list of Databases in EM.

Name	Type	Status	Target Version	CPU	Memory	Disk	Operating System
cdb186.subnet.vcn.oraclevcn.com	Database Inst...	▲	18.8.0.0.0	0	0	0	N/A
emrep.us.oracle.com	Database Inst...	▲	19.3.0.0.0	0	0	0	N/A
hr.subnet.vcn.oraclevcn.com	Database Inst...	▲	18.3.0.0.0	0	0	0	N/A
sales.subnet.vcn.oraclevcn.com	Database Inst...	▲	18.3.0.0.0	0	0	0	N/A

Here you will notice different databases listed, such as SALES, HR etc, we will work the sales container database

1.3 Select Sales database from the list, this will take you to the DB home page for this database

ORACLE ENTERPRISE MANAGER Cloud Control 13c

Database Management Lab

Databases

Name	Type	Status	Target Version	Incidents	Average Compliance Score	Member Status Summary	Operating System
cdb186.subnet.vcn.oraclevcn.com	Database Inst...	▲	18.8.0.0.0	0 0 0	N/A	0 2 0 0	Linux
emrep.us.oracle.com	Database Inst...	▲	19.3.0.0.0	0 0 0	N/A	0 0 0 0	Linux
hr.subnet.vcn.oraclevcn.com	Database Inst...	▲	18.3.0.0.0	0 0 0	N/A	0 1 0 0	Linux
sales subnet vcn.oraclevcn.com	Database Inst...	▲	18.3.0.0.0	0 0 0	N/A	0 5 0 0	Linux

Deprecated Search List (with Metrics)

sales.subnet.vcn.oraclevcn.com (Container Database...)

18.3.0.0.0 Version

0 Application Roots 5 (▲5) Pluggable Databases 0 days, 15 hrs Up Time 55.2% Availability for Last 7 Days N/A Last Backup

Load and Capacity

0.46 Average Active Sessions 11.64 Used Space (GB)

Incidents and Compliance

0 0 1 0 Compliance Not Configured

Recommendations

0 ADDM Findings

Performance

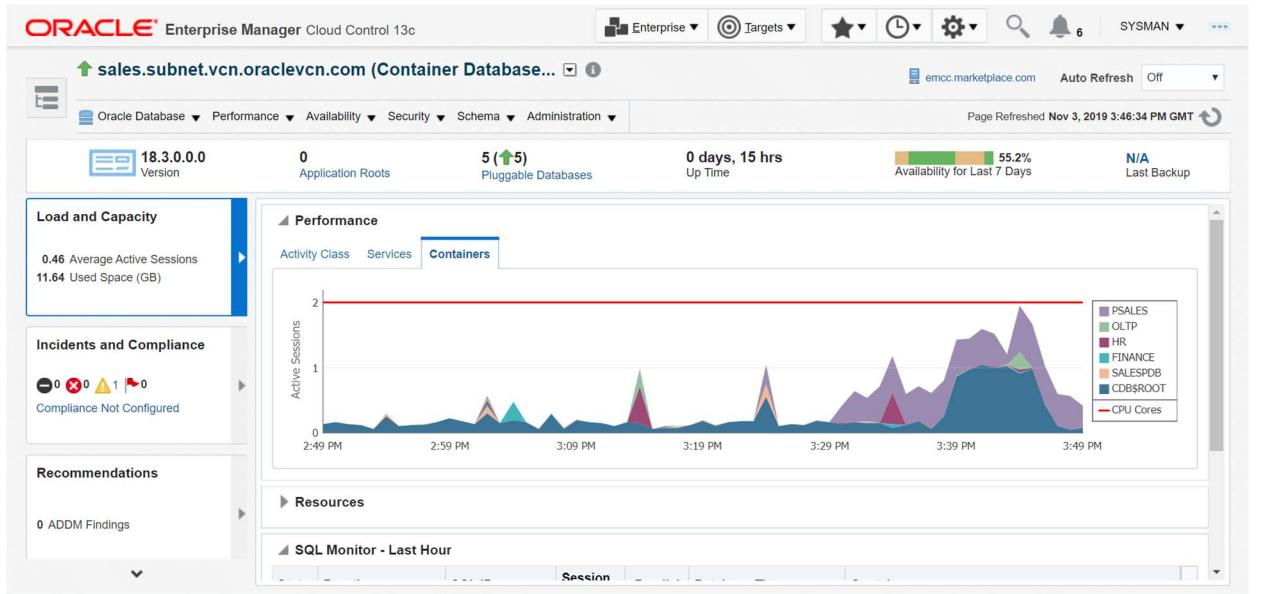
Activity Class Services Containers

Active Sessions

Resources

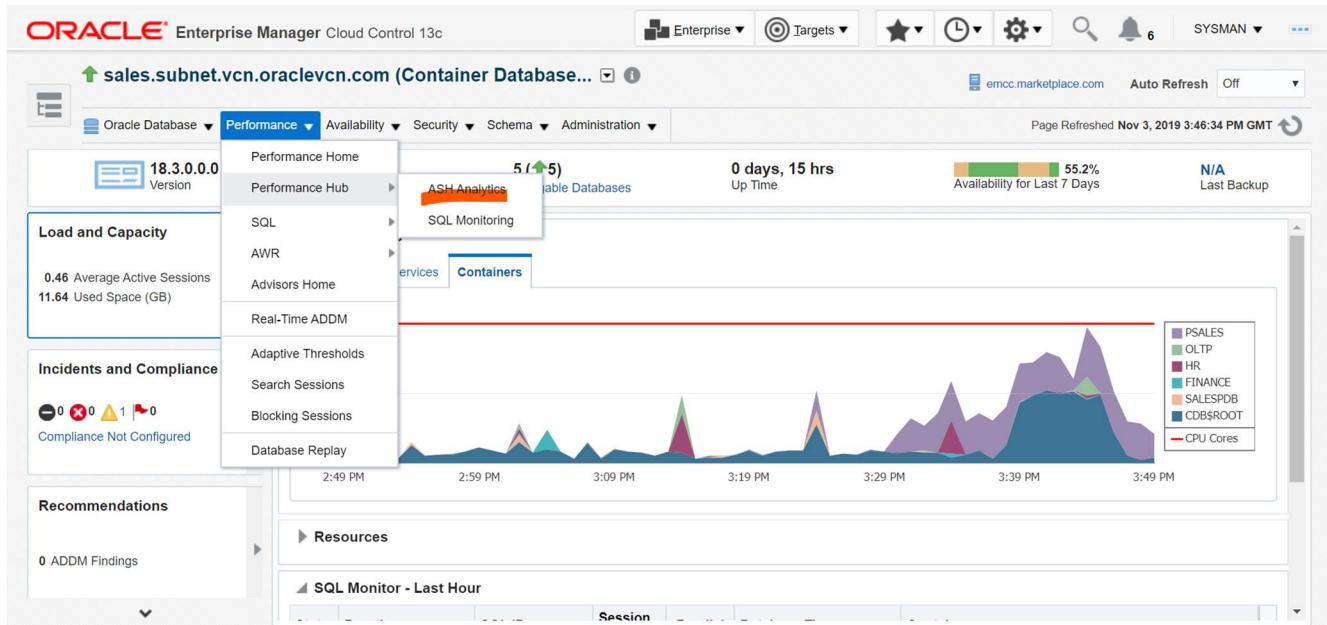
SQL Monitor - Last Hour

- Click on the Containers tab. It is located at the upper right-hand corner of the page, underneath the Performance tile. This will show the list of pluggable databases in the CDB and their activity



Notice that the PSALES database is the busiest. We focus our attention to this PDB. Let us now navigate to Performance Hub

- 1.5 Select Performance Hub from the Performance Menu and click on ASH Analytics and use the sales_system credential name from the database login screen



ORACLE ENTERPRISE MANAGER Cloud Control 13c

Database Management Lab

The screenshot shows two main sections of the Oracle Enterprise Manager interface.

Database Login: This section is titled "Database Login" and shows a "Credential" configuration. It has a "Named" radio button selected, and the "Credential Name" is set to "SALES_SYSTEM". Below this, there is a table titled "Credential Details" with three rows: Username (system), Password (*****), and Role (normal). At the bottom are "Login" and "Cancel" buttons.

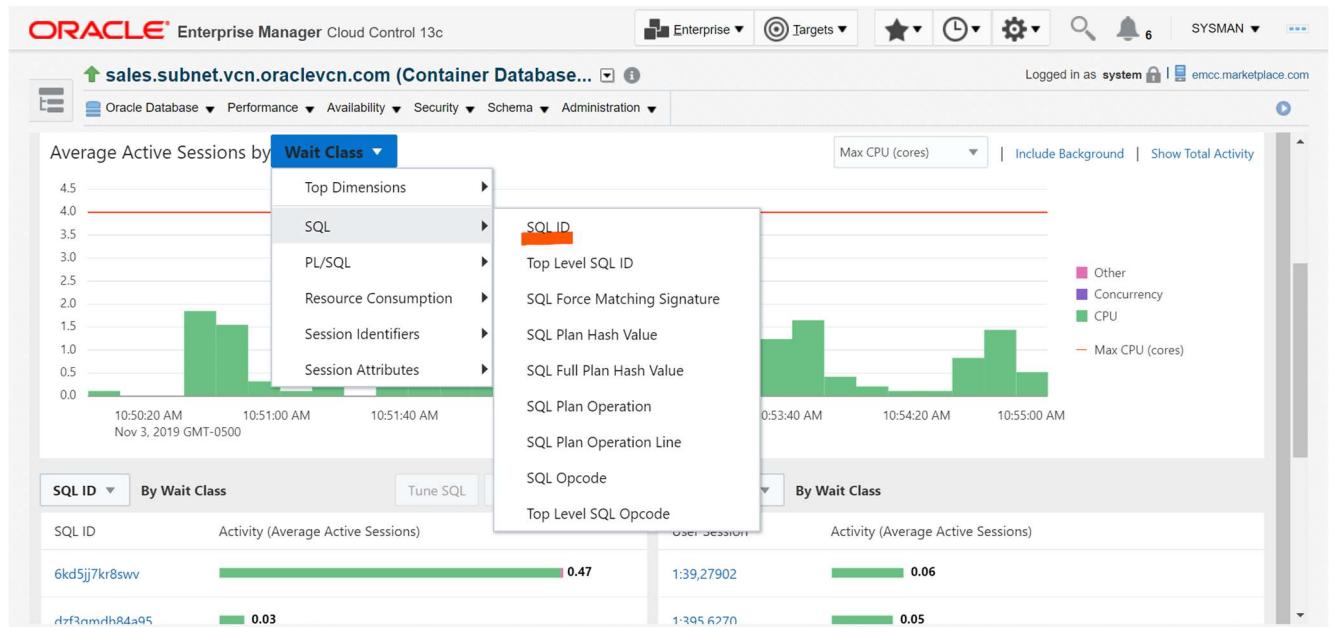
Performance Hub: This section is titled "Performance Hub" and displays a performance summary. It includes a graph showing CPU usage over time (Nov 3, 2019 GMT-0500) with a green area representing CPU activity. A legend indicates Wait (orange), User I/O (blue), and CPU (green). Below the graph, there are tabs for "ASH Analytics" (selected) and "SQL Monitoring". The "ASH Analytics" tab shows a chart for "Average Active Sessions by Wait Class" with a red line at 4.0 and a green bar at 2.0. The right side of the Performance Hub shows a summary of activity with a "Last hour" report from Nov 3, 2019, 09:54 AM to 10:54 AM.

Make sure to slide the time picker on an area of high usage (e.g., CPU, IO or Waits). Notice how the corresponding selected time window also changes in the summary section.

You can also resize the slider to entirely cover the time period of your interest. Notice the graph at bottom, it is providing more detailed view of the time window you selected. By Default the wait class dimension is selected. On the right hand side of the graph you have a list of wait classes for the time window you selected (blue for user I/O, green for CPU etc). Notice how the color changes if you hover over either the menu or the graph to highlight the particular wait class.

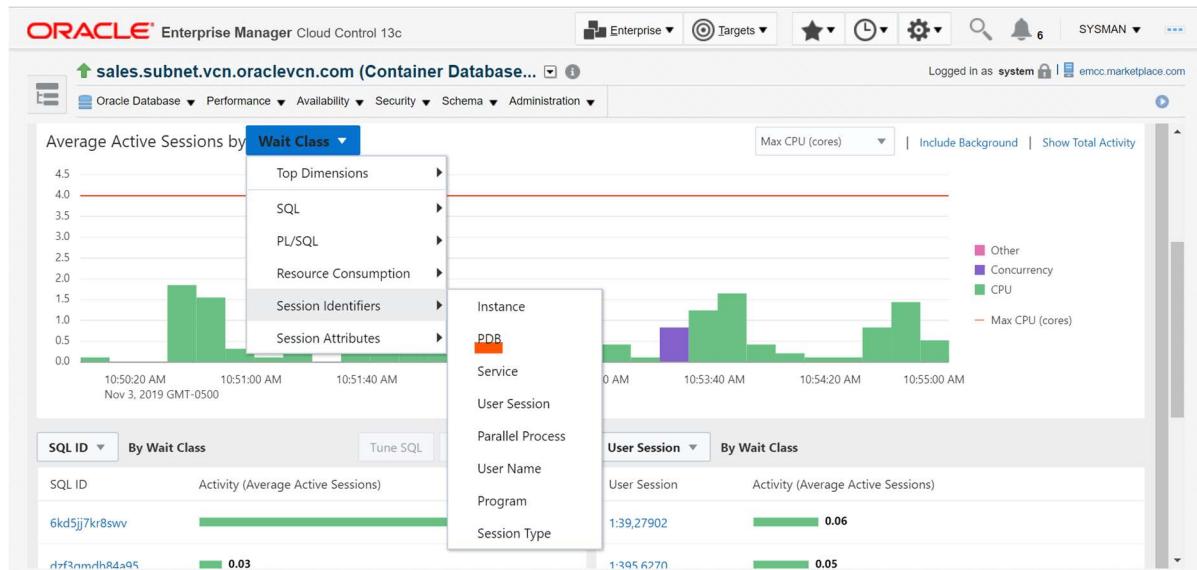
Wait class isn't the only dimension you can drill into the performance issue by. Let's say you wanted to identify the SQL that was causing the biggest performance impact. You can do that by clicking the drop down list and changing the top dimension from wait class to SQL ID.

1.6 Select the SQL ID dimension from the list of available dimensions(Under Top Dimensions) using the dropdown box that is currently displaying Wait Class. Top Dimensions → SQL ID



Hover your mouse on top of the SQL (one at the bottom) and you will be able to see how much activity is consumed by this SQL.

Now using the same list of filters select the PDB dimension. Session Identifiers → PDB

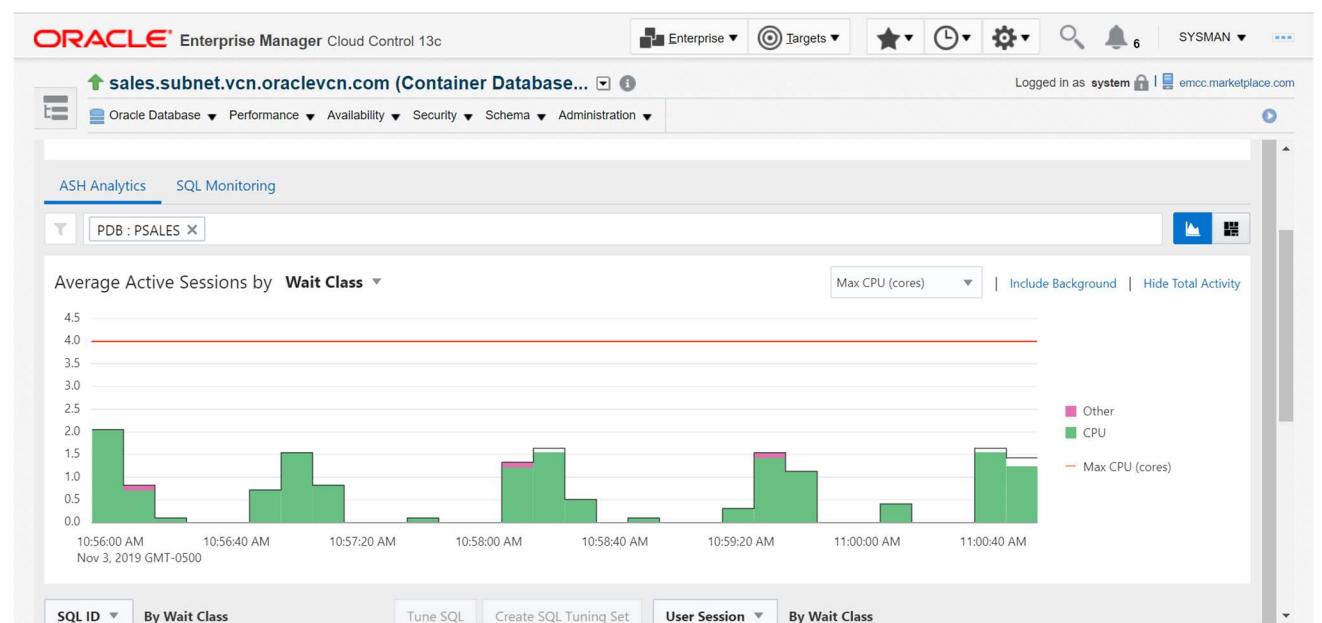
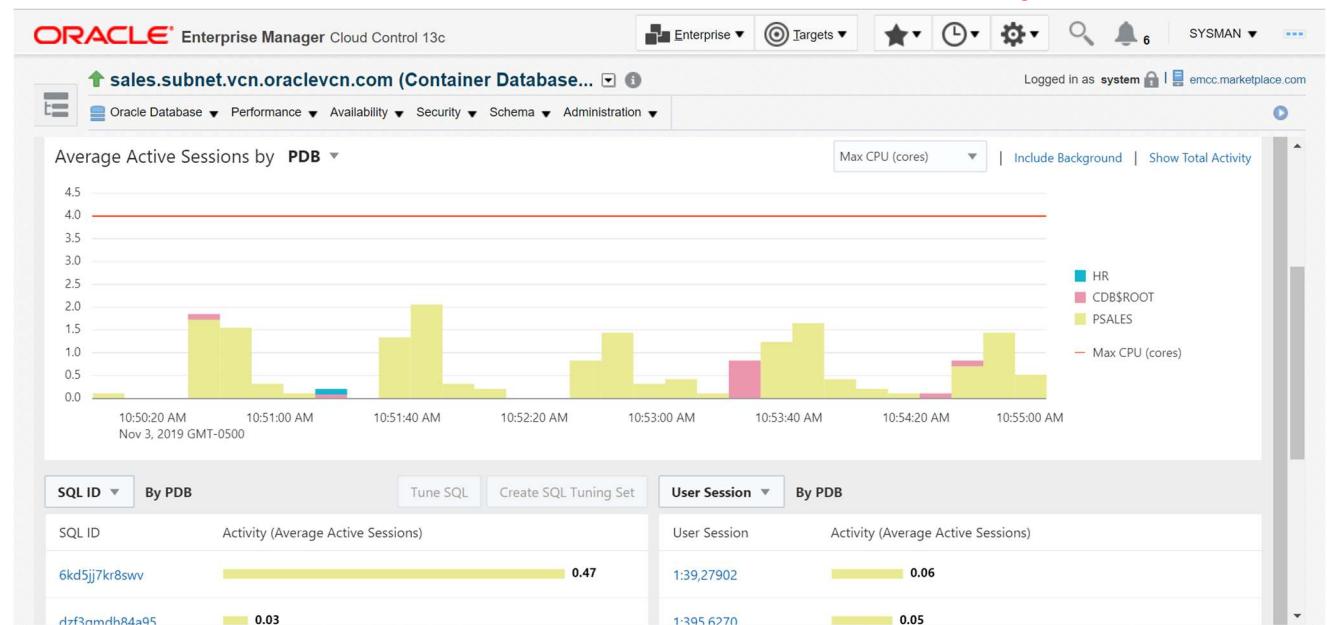


What do you see?

The chart changes to activity by the different pluggable databases created in this Container database. Click on the 'PSALES' pluggable database on the list to add it to the filter by list and drilldown to activity by this PDB on the same page.

ORACLE ENTERPRISE MANAGER Cloud Control 13c

Database Management Lab



1.7 Click on the SQL Monitoring Tab

Status	Duration	SQL ID	SQL Plan Hash	User Name	Parallel	Database Time	I/O Requests	SQL Text
●	1.00s	6kd5jj7kr8swv	2123671122	SH2@PSALES	+4	0.01s		SELECT MAX(AMOUNT_SOLD) FROM SALES WHERE CUST_ID IN (SELECT CUST_ID FROM SH2@PSALES.CUSTOMERS WHERE CUST_NAME = 'ABC') ORDER BY 1
●	16.00s	3j3t587bt0vqw	0	SH2@PSALES		9.74s		declare -- v1 varc
✓	1.00s	6kd5jj7kr8swv	2123671122	SH2@PSALES	+4	0.74s		SELECT MAX(AMOUNT_SOLD) FROM SALES WHERE CUST_ID IN (SELECT CUST_ID FROM SH2@PSALES.CUSTOMERS WHERE CUST_NAME = 'ABC') ORDER BY 1
✓	1.00s	6kd5jj7kr8swv	2123671122	SH2@PSALES	+4	0.54s		SELECT MAX(AMOUNT_SOLD) FROM SALES WHERE CUST_ID IN (SELECT CUST_ID FROM SH2@PSALES.CUSTOMERS WHERE CUST_NAME = 'ABC') ORDER BY 1
✓	0.49s	6kd5jj7kr8swv	2123671122	SH2@PSALES	+4	0.49s		SELECT MAX(AMOUNT_SOLD) FROM SALES WHERE CUST_ID IN (SELECT CUST_ID FROM SH2@PSALES.CUSTOMERS WHERE CUST_NAME = 'ABC') ORDER BY 1

You can see all the executed SQL during that time along with different attributes like ‘user’, ‘Start’, ‘Ended’ etc. The test next to the @ sign indicates the name of the PDB. Click on any SQL of your choice (e.g. 6kd5jj7kr8swv)

It will navigate you to show the details of this particular query. You can see the plan, parallelism and activity of the query. “Plan Statistics” tab is selected by default. You can see the plan of this query in graphical mode.

In some cases, the Monitored SQL may have aged out and no rows are displayed, in this case try using the time-picker and pick last 24 hrs time period to identify the historical SQL that was monitored. This is an issue being investigated.

- 1.7.1 Select “Parallel” tab. This will give details about parallel coordinator and parallel slaves.
- 1.7.2 Click on the SQL Text tab. You can see the query text which got executed.
- 1.7.3 Click on the activity tab to understand about the activity breakdown for this SQL

- 1.7.4** Click on “Save” button on top right corner of the page. This will help you to save this monitored execution in “.html” format, which you can use it to share or to diagnose the things offline.

B. Real-Time Database Operations Monitoring

Environment Details:

The target database is running a load on the sample schema provided via the Examples (or companion) software accompanying the Oracle database software. There are other schemas created to simulate specific performance scenarios.

Estimated Time to Complete Use Case: **8 minutes**

Business Case

Real-Time Database Operations Monitoring, feature introduced in Oracle database 12c, allows an administrator to monitor long running database tasks such as batch jobs, ETLs etc as a composite business operation. This feature tracks the progress of SQL and PL/SQL queries associated with the business operation being monitored. Developers and DBAs can define business operations for monitoring by explicitly specifying the start and end of operation or implicitly through the use of tags that identify the operation.

Start the Database Operation.

- ssh using a terminal window or putty and log in as oracle. ([see section above: Lab environment setup](#))
- Once Logged in perform the following

```
[oracle@em12 ~]$ cd scripts  
[oracle@em12 scripts]$ ./SALEENV  
[oracle@em12 scripts]$ cd load/frame/queries/awrv  
[oracle@em12 awrv]$ pwd  
/home/oracle/scripts/load/frame/queries/awrv
```

Using SQLPlus connect to the sh2 account.

```
[oracle@em12 awrv]$ sqlplus sh2/sh2@psales
```

Open the file (!vi DBOP.sql) from the SQL prompt and then review the content of the file.

At the beginning of the file you will notice how we have tagged the operation with dbms_sql_monitor.begin_operation and ended it with dbms_sql_monitor.end_operation

Now execute the file @DBOP.sql

2.1 You should already be logged on to Enterprise Manager. If you are not, please follow the instructions detailed in earlier section of this workbook.

2.2 Select the Monitored SQL tab.

Review the list of currently executing SQLs are visible click on the DBOP_DEMO name. This will open the DBOP named DBOP_DEMO.

Note: You may need to scroll down or select “Database operations” from the type dropdown.

The screenshot shows the Performance Hub section of the Oracle Enterprise Manager interface. At the top, there's a chart showing CPU usage over time from 10:25 AM to 11:20 AM on Nov 3, 2019. The legend indicates Wait (orange), User I/O (blue), and CPU (green). Below the chart, the "Top 100 By Last Active Time" section displays two SQL queries:

Status	Duration	SQL ID	SQL Plan Hash	User Name	Parallel	Database Time	I/O Requests
●	19.00s	DBOP_DEMO		SH2@PSALES		18.73s	4,316
●	0.46s	153yacm7ht8nd	96485341	SH2@PSALES	+4	0.46s	

2.3 Review the details of the Database Operations.

The screenshot shows the "Execution Details" page for execution ID 2. It includes sections for Overview, General, Time & Wait Statistics, I/O Statistics, and Activity. The Activity tab is selected, showing a stacked bar chart of CPU usage by SQL ID. The General section provides details like Execution Plan, Start Time, and User Name.

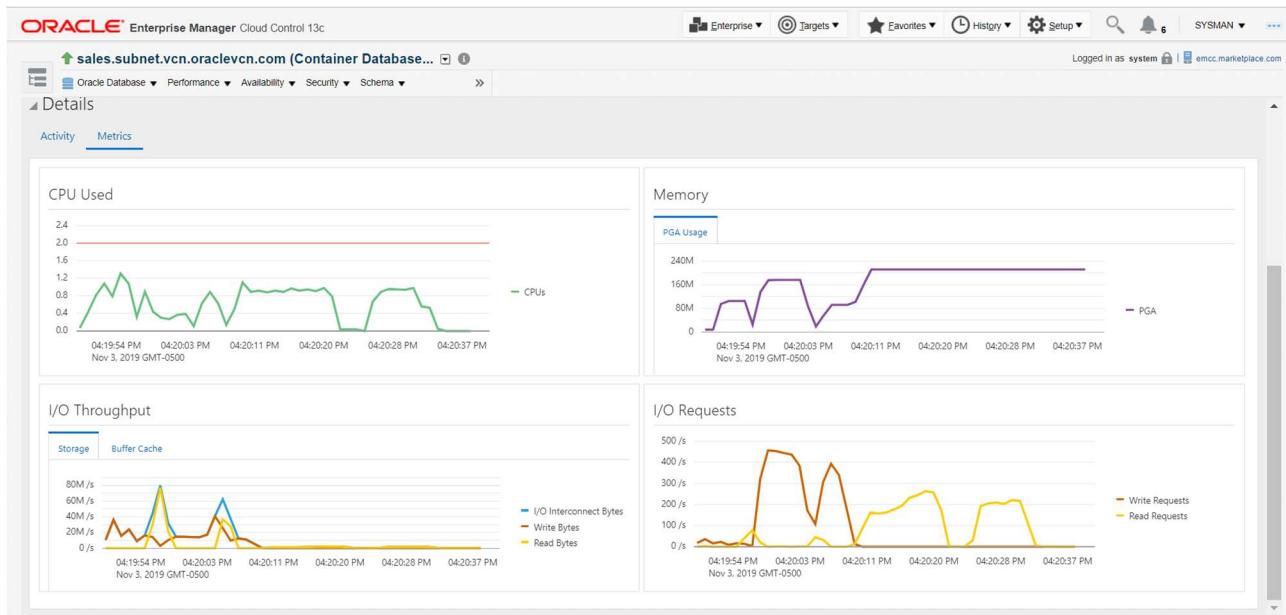
Execution Plan	DBOP_DBOP_DEMO
Execution Started	11/3/2019, 4:19:50 PM
Last Refresh Time	11/3/2019, 4:20:39 PM
Execution Id	2
User Name	SH2@PSALES

Duration	49.0s
Database Time	48.8s
PL/SQL & Java	0s
Activity %	100%

Buffer Gets	70K
I/O Requests	8,040
I/O Bytes	517MB

2.4 Click on the Activity tab.

You will see all the activity for this operation.



C. Tuning a SQL in a PDB

Environment Details:

The target database is running a load on the sample schema provided via the Examples (or companion) software accompanying the Oracle database software. There are other schemas created to simulate specific performance scenarios.

Lab environment setup

Your environment should have a workload running. If EM 13c is not running please refer to Appendix B for the steps to start EM 13c.

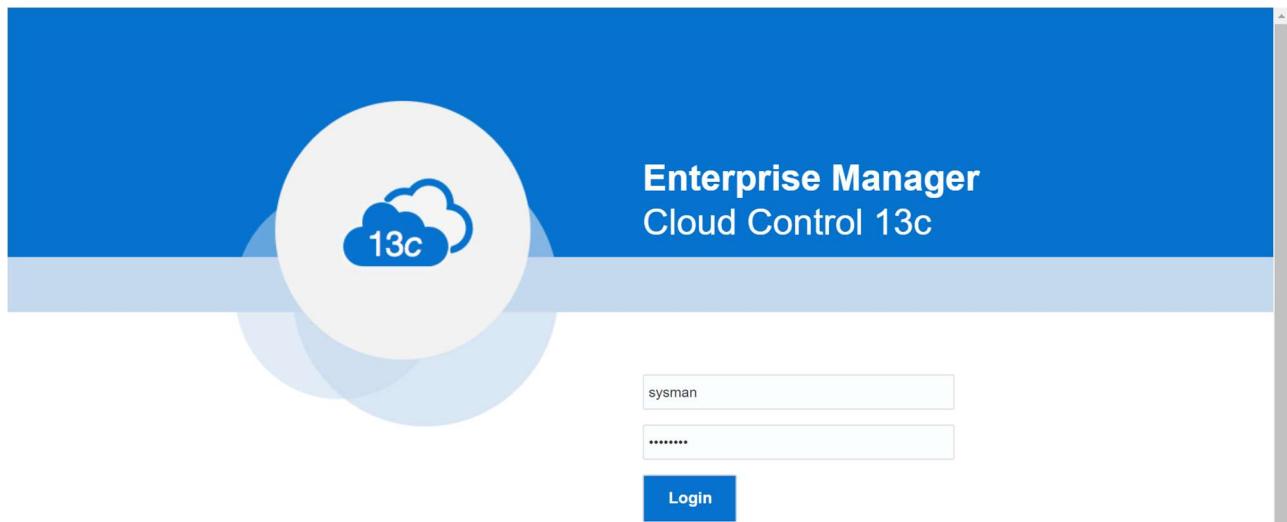
Estimated Time to Complete Use Case: **10 minutes**

Business Case

This use case is intended to give an idea of how the pluggable database administrator will tune queries in a PDB. We are running a workload and this flow will help you to identify a Top SQL and then tune it using SQL Tuning Advisor. The PDPA will have no access to the Container and his/her view is only restricted to the queries running in the PDB assigned to this user.

- 3.1 Log into Enterprise Manager Cloud Control.
- 3.2 Please note that you will be using a new URL: https://Your_Assign_IP:7803/em

Login: sysman/welcome1



ORACLE ENTERPRISE MANAGER Cloud Control 13c

Database Management Lab

- 3.3 Once logged into Enterprise Manager. Select Targets → Databases . Click on the expand icon on the left and click on the database sales.subnet.vcn.oraclevcn.com

The screenshot shows the 'Databases' section of the Oracle Enterprise Manager interface. The top navigation bar includes links for Performance, Availability, Security, Schema, and Administration. The main content area displays a table of databases with columns for Name, Type, Status, Target Version, Incidents, Average Compliance Score, Member Status Summary, Operating System, Platform, and Site. The database 'sales.subnet.vcn.oraclevcn.com' is listed and has its row highlighted in orange, indicating it is selected.

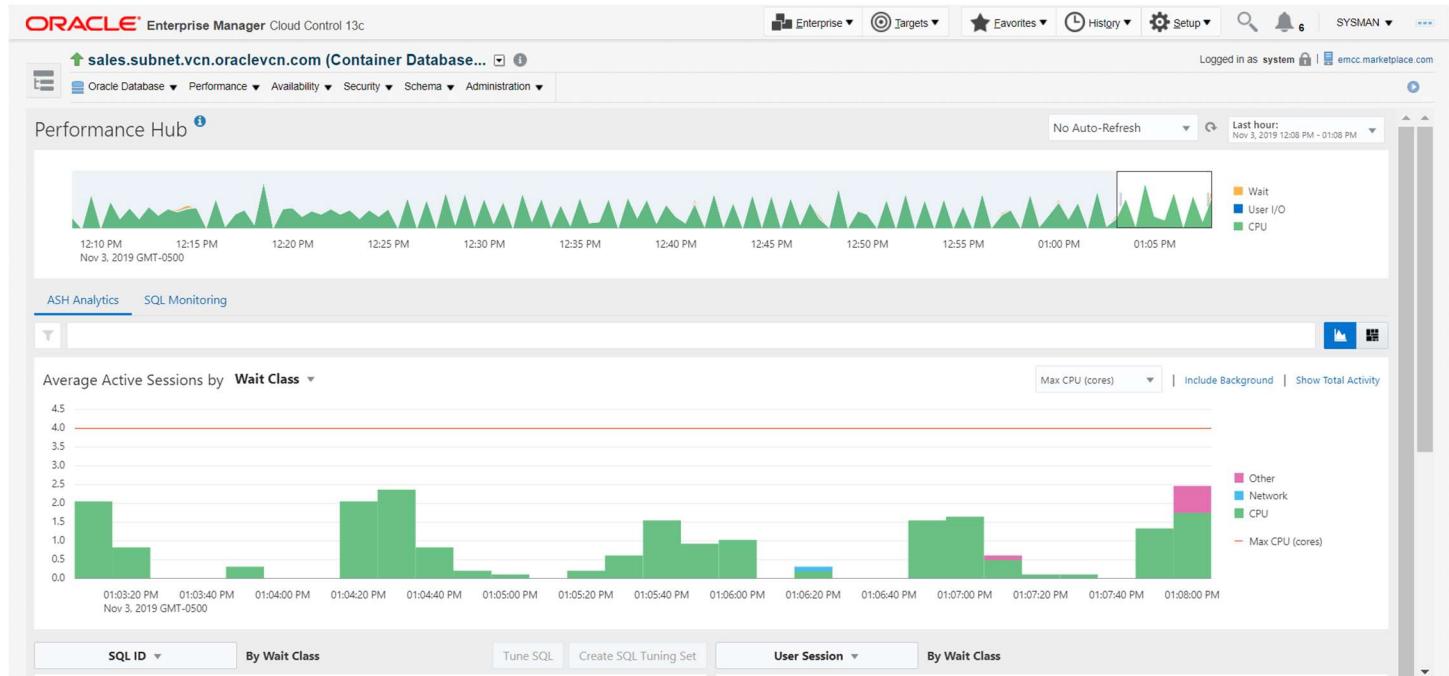
- 3.4 You should now see the Database Home page.

The screenshot shows the Database Home page for the 'sales.subnet.vcn.oraclevcn.com' database. The top header shows the database name and a container status. The main dashboard includes sections for Load and Capacity (showing 18.3.0.0 version, 0.60 average active sessions, and 11.64 used space), Incidents and Compliance (showing 0 incidents and 1 compliance issue), Recommendations (1 ADDM findings), High Availability (N/A last backup status, Data Guard not configured), and Jobs. The central area features a Performance timeline chart showing active sessions over time, categorized by activity class (PSALES, OLTP, HR, FINANCE, SALESPOB, CDB\$ROOT, CPU Cores). Below the chart is a SQL Monitor table for the last hour, listing various SQL statements with their status, duration, session ID, parallel degree, database time, and container.

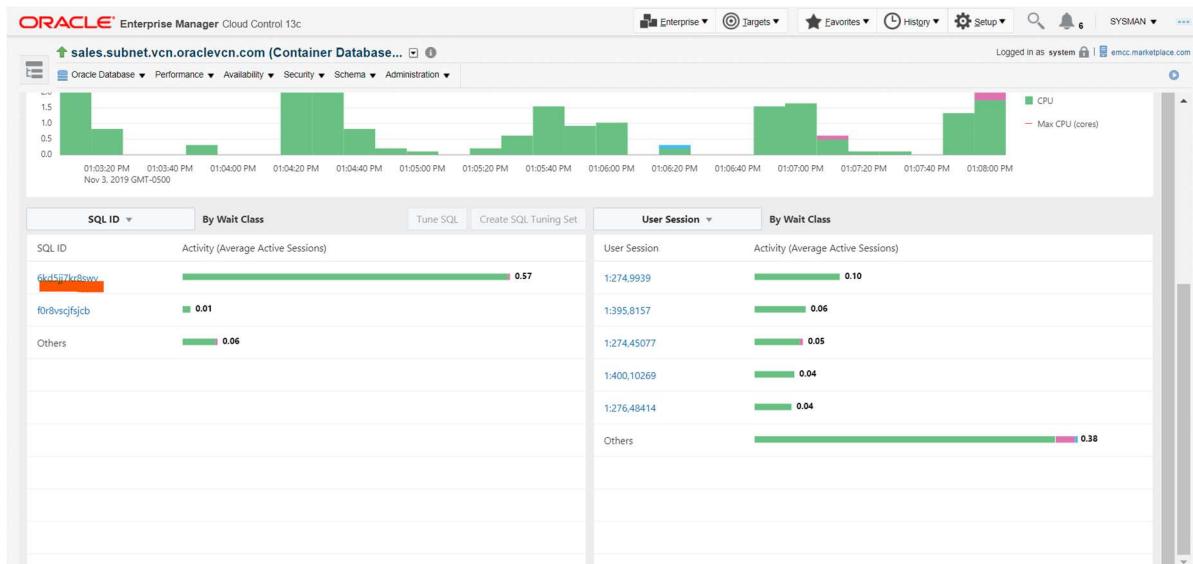
- 3.5 From the Performance Menu click on Performance Hub-> ASH Analytics

ORACLE ENTERPRISE MANAGER Cloud Control 13c

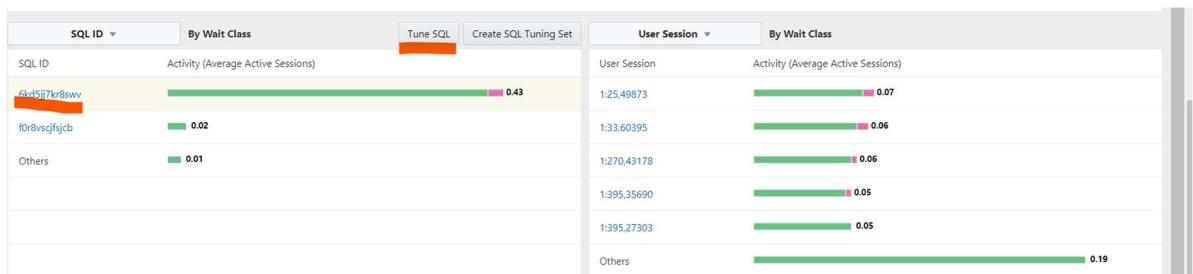
Database Management Lab



3.6 Click on the activity bar for the SQL showing highest activity in the bottom left region.



3.7 Now schedule the SQL Tuning Advisor by clicking on the 'Tune SQL' button.



3.8 Accept the default and submit the SQL tuning Job.

The screenshot shows the 'Schedule SQL Tuning Advisor' page. Key fields include:

- Name:** SQL_TUNING_1572804547929
- SQL Tuning Set:** SYSTEM.TOP_ACTIVITY_1572804547007
- Scope:** Total Time Limit (minutes): 30; Scope of Analysis: Comprehensive (selected), Time Limit per Statement (minutes): 5
- Schedule:** Time Zone: UTC+00:00 Coordinated Universal Time (UTC); Date: Nov 3, 2019; Time: 06:09:00 AM

The screenshot shows the processing status of the SQL Tuning Advisor Task. Key details include:

- Task Name:** SQL Tuning Advisor Task SQL_TUNING_1572804547929
- Status:** INITIAL
- Elapsed Time (seconds):** 0
- Task Progress:**
 - Creating a new SQL Tuning set (✓)
 - Creating a new SQL Tuning task (✓)
 - Executing the task (➡)

3.9 Once the job completes. You should see the recommendations for either creating a profile or an index.

ORACLE ENTERPRISE MANAGER Cloud Control 13c

Database Management Lab

Only one recommendation should be implemented.

SQL Information

Parsing Schema SH2
PDB Name SALES
SQL Text SELECT MAX(AMOUNT_SOLD) FROM SALES WHERE CUST_ID IN (SELECT CUST_ID FROM CUSTOMERS WHERE CUST_CITY=:B1)

Select Recommendation

Original Explain Plan (Annotated)

Implement Validate with SPA

Select	Type	Findings	Recommendations	Rationale	Benefit (%)	Other Statistics	New Explain Plan	Compare Explain Plans
<input checked="" type="radio"/>	SQL Profile	A potentially better execution plan was found for this statement.	Consider accepting the recommended SQL profile. No SQL profile currently exists for this recommendation.	Creating the recommended indices significantly improves the execution plan of this statement. However, it might be preferable to run "Access Advisor" using a representative SQL workload as opposed to a single statement. This will allow to get comprehensive index recommendations which takes into account index maintenance overhead and additional space consumption.	52.81			
<input type="radio"/>	Index	The execution plan of this statement can be improved by creating one or more indices.	Consider running the Access Advisor to improve the physical schema design or creating the recommended index SH2.CUSTOMERS(CUST_CITY, CUST_ID)		76.51			

- 3.10 Implement the SQL Profile recommendation. SQL Profiles are a great way of tuning a SQL without creating any new objects or making any code changes.

At this point let's now turn off the load: Change directory to scripts and execute the script 1-db_lab_stop.sh as shown below

This concludes the Database Performance Management lab. You can now move on to Real Application Testing lab.

Real Application Testing

Objective:

The objective of this lab to provide exercises designed to showcase the new Real Application Testing capabilities in Oracle Enterprise Manager Cloud Control 13c and Database 18c.

Functional Coverage:

In this lab you will go through features in the following functional areas:

- **SQL Performance Analyzer Optimizer Statistics**

[SQL Performance Analyzer Gather Optimizer Statistics Validation](#)

- **Consolidated Database Replay**

[Replay multiple workloads concurrently against Pluggable Databases in the Container Database](#)

- **Appendix A**

Exercise overview and Business Case

D. SQL Performance Analyzer Optimizer Statistics

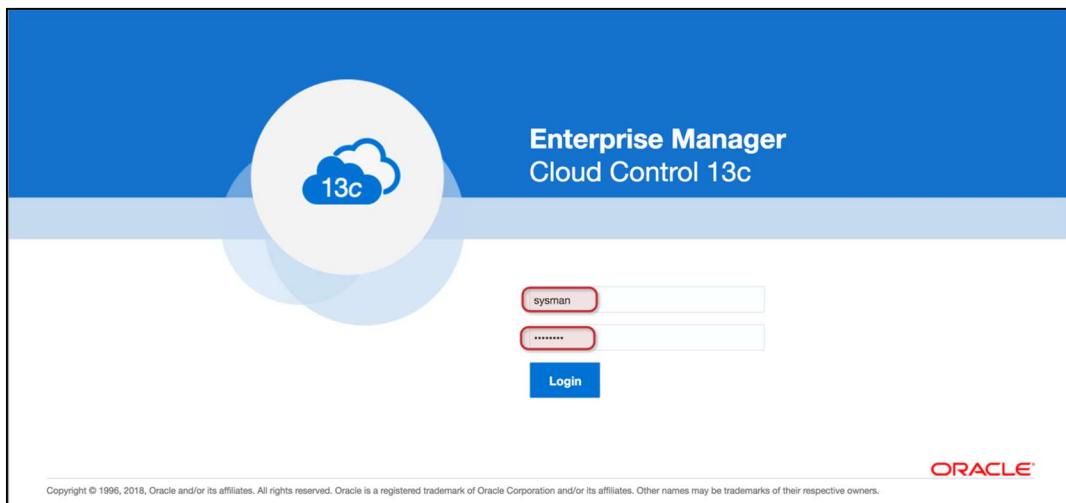
SPA Lab Workflow

Estimated Time to Complete Use Case: **9 minutes, Recommended workflow**

In this exercise we need to configure the database to set up optimizer statistics to be stale. So first step is to create and submit a job that will configure the statistics to be stale.

4. Execute SPA task using Optimizer statistics workflow

4.1 Login using username and password sysman/ welcome1



4.2 Navigate to Job library: From Enterprise > Job > Library

4.3 Select “OS Command” in the Create library Job drop down list

Click Go

Job Type	Owner	Targets	Target Type
SQL Script	SYS	emcc.marketplace.com	Database Instance
OS Command	SYSMAN		Host

4.4 Provide Name: STAT_SETUP

Click Add in Target section

Select “emcc.marketplace.com”

Select Tab “Parameters”

Name	Type	Host	Time Zone
emcc.marketplace.com	Host	emcc.marketplace.com	Greenwich Mean Time

4.5 In Command enter: "/home/oracle/scripts/dbpack/dbpacks_scripts/demo/SPA_STATS_SETUP19"

The screenshot shows the 'Job' creation interface. The 'Command' field is highlighted with a red box and contains the value '/home/oracle/scripts/demo/SPA_STATS_SETUP19'. The 'Target Properties' section on the right lists several environment variables with their descriptions:

- %emd_root% location of Agent
- %perfbin% location of Perf binary used by Agent
- %TargetName% target name
- %TargetType% target type
- %SNMPHostname% SNMP Hostname
- %Timeout% SNMP Timeout (Default: 10 seconds)
- %WBEMPort% Port number for WBEM Access Default: 5988
- %LIMIT_TOP% Disk Activity Metrics Collection Max Rows Upload(>0) Default:16
- %NMUPM_SUPPORT_LOFS% Monitor Loopback Filesystems (true/false) Default:false
- %NMUPM_USE_PSEUDO_MEM% Use pseudo-memory for Swap utilization (true/false) Default:true
- %RemoteCacheRoot% This directory serves as root for remote cache
- %RemoteCacheSbinRoot% This directory serves as root of sbin content for remote cache

4.6 Select Tab "Credentials"

Select "Named"

Select "ORACLE_HOST"

Click "Save to Library"

The screenshot shows the 'Edit OS Command Library Job: STAT_SETUP' screen. The 'Credentials' tab is selected. A 'Credential Name' dropdown is highlighted with a red box and set to 'ORACLE_HOST'. Below it, 'Credential Details' show 'UserName' as 'oracle' and 'Password' as '*****'.

4.7 Select "STAT_SETUP"

Click "Submit"

The screenshot shows the 'Job Library' screen. The 'STAT_SETUP' job is selected and highlighted with a red box. The table below lists other jobs:

	Job Type	Owner	Targets	Target Type
DISABLE TABLESPACE USED (%) ALERTS FOR UNDO AND TEMP TABLESPACES	SQL Script	SYS	Database Instance	
START_DB_PERF_WORKLOAD	OS Command	SYSMAN	emcc.marketplace.com	Host
STAT_SETUP	OS Command	SYSMAN	emcc.marketplace.com	Host

4.8

Click "Submit"

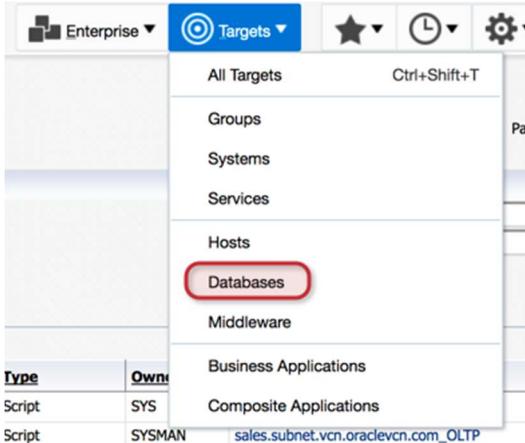
The screenshot shows the 'Job' configuration page for a job named 'STAT_SETUP1'. The 'General' tab is selected. The 'Name' field contains 'STAT_SETUP1'. Under 'Automatic Attempts', the dropdown is set to '<none>'. The 'Target Type' is set to 'Host'. The 'Target' section lists a single target: 'emcc.marketplace.com' (Type: Host, Hostname: emcc.marketplace.com, Time Zone: Greenwich Mean Time). The 'Submit' button at the top right is highlighted with a red box.

The job is now running

Continue with configuring SPA Quick Check

4.9

Navigate to Databases: From the menu, Targets -> Databases



4.10 Expand the sales.subnet.vcn.oraclevcn.com database

Click on “sales.subnet.vcn.oraclevcn.com_HR” pluggable database

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface under the 'Databases' section. The navigation bar at the top includes 'Enterprise', 'Targets', 'SYSMAN', and other system status indicators. Below the navigation is a search bar and a toolbar with 'View', 'Add', 'Remove', and 'Configure' buttons. The main area displays a table of databases with columns: Name, Type, Status, Target Version, Incidents, Average Compliance Score, Member Status Summary, Operating System, and Platform. Several databases are listed, including 'cdb186.subnet.vcn.oraclevcn.com', 'emrep.us.oracle.com', 'hr.subnet.vcn.oraclevcn.com', and 'sales.subnet.vcn.oraclevcn.com'. Under 'sales.subnet.vcn.oraclevcn.com', there are four pluggable databases: 'sales.subnet.vcn.oraclevcn.com_FINANCE', 'sales.subnet.vcn.oraclevcn.com_HR' (which is highlighted with a red box), 'sales.subnet.vcn.oraclevcn.com_OLTP', and 'sales.subnet.vcn.oraclevcn.com_PSALES'. The 'Platform' column indicates all are x86_64.

4.11 In sales.subnet.vcn.oraclevcn.com_HR database Navigate to Performance -> SQL -> SQL Performance Analyzer Quick Check Setup

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface for the 'sales.subnet.vcn.oraclevcn.com / HR' database. The left sidebar has sections for 'Load and Capacity', 'Incidents and Compliance', and 'Jobs'. The main area is titled 'Performance Hub' and shows a timeline from 8:06 AM to 9:05 AM. It includes a chart for 'Availability for Last 7 Days' (83.36%) and a 'SQL Monitor - Last Hour' section with a table for 'Status Duration', 'SQL ID', 'Session ID', 'Parallel', and 'Database Time'. A vertical menu on the left under 'Performance' has options like 'SQL', 'AWR', 'Advisors Home', 'Search Sessions', and 'Blocking Sessions'. The 'SQL' option is highlighted with a red box, and 'SQL Performance Analyzer Quick Check Setup' is also highlighted with a red box within the 'SQL' menu options.

- 4.12** This is the page where you configure SPA Quick Check. Make sure that the selected SQL Tuning Set includes as many SQL statements as possible. If the application has specific workloads that are executed during End of Month, End of Year or even certain period during the day, then make sure to collect the workload in separate SQL Tuning Sets and merge them into a “Total Workload Tuning set”

In this example we are working with a SQL Tuning Set called “PENDING_STATS_WKLD”

Select: SQL Tuning Set : PENDING_STATS_WKLD

Select “Comparision Metric” : Buffer Gets

Click “Save”.

The screenshot shows the 'SPA Quick Check Setup' page in Oracle Enterprise Manager. The 'SQL Tuning Set' dropdown is set to 'SYSTEM.PENDING_STATS_WKLD'. The 'Comparison Metric' dropdown is set to 'Buffer Gets'. The 'Save' button at the bottom is highlighted with a red box.

4.13 Go To Performance > SQL > Optimizer Statistics

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface. The top navigation bar includes links for Enterprise, Targets, Home, Database, Performance, Availability, Security, Schema, Administration, and SYSMAN. A confirmation message in the top left corner states "Settings are saved successfully". The main content area is titled "sales.subnet.vcn.oraclevcn.com / HR". The left sidebar has a "SPA Quick Check Setup" section and a "SQL Performance Advisor" section. The "SQL Performance Advisor" section contains several configuration options and a sidebar with links like "SQL Tuning Advisor", "SQL Performance Analyzer Home", "SQL Plan Control", and "Optimizer Statistics". The "Optimizer Statistics" link is highlighted with a red box. The right side of the screen displays detailed information about optimizer statistics, including a "SQL Impact Threshold(%):" section and a "Multiple executions:" section.

4.14 Click "Gather"

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface. The top navigation bar is identical to the previous screenshot. The main content area is titled "Optimizer Statistics Console". The left sidebar has sections for "Operations" (with "Gather" highlighted with a red box), "Statistics Status" (showing a pie chart with values 1,589, 18, and 2,602), and "Statistics Gathering Job List". The right side of the screen has sections for "Configure" (with "Global Statistics Gathering Options" and "Object Level Statistics Gathering Options"), "Statistics Gathering Auto Tasks" (a progress bar chart showing task completion percentages from 0% to 100%), and "Statistics Gathering Job List" (a table with columns: Operation Name, Job Name, Target, Status, Total Objects, Num Completed, Start Time, Duration).

4.15 Select "Schema"

Check "Validate the impact of statistics on...."

Click "Next"

The screenshot shows the 'Gather Optimizer Statistics: Scope' step of a process. Under 'Object Type', the 'Schema' radio button is selected (highlighted with a red box). In the 'Validate with SQL Performance Analyzer' section, the checkbox 'Validate the impact of statistics on SQL performance before publishing (recommended)' is checked (also highlighted with a red box).

4.16 Click "Add"

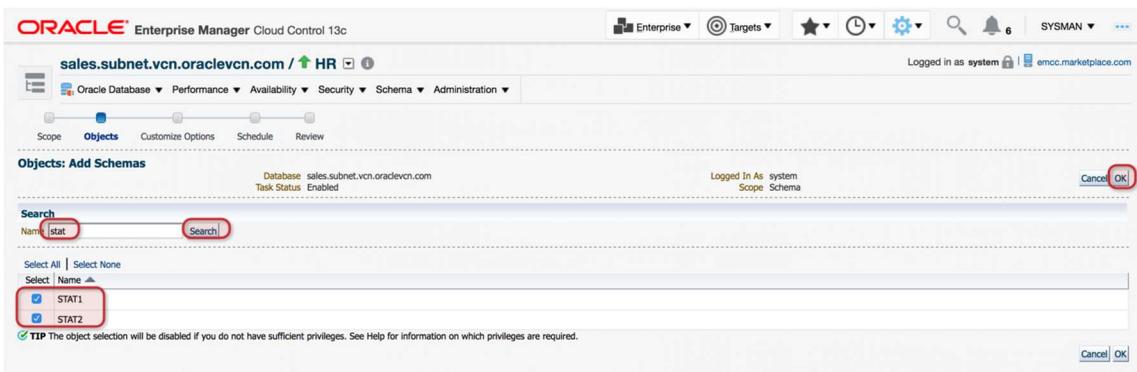
The screenshot shows the 'Gather Optimizer Statistics: Objects' step. At the bottom right, there is an 'Add' button (highlighted with a red box).

4.17 Enter "Name": STAT

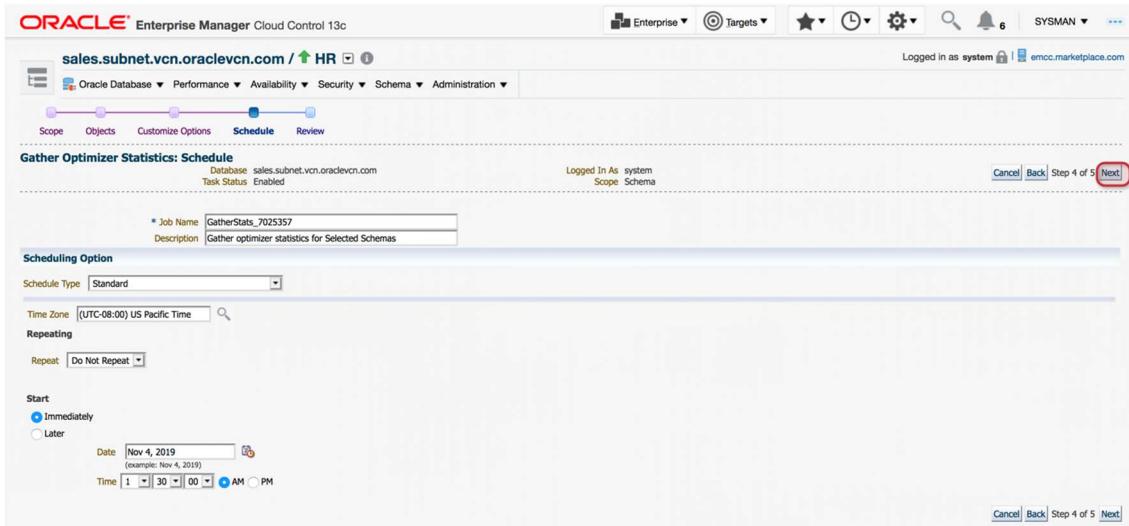
Click "Search"

Select: STAT1, STAT2

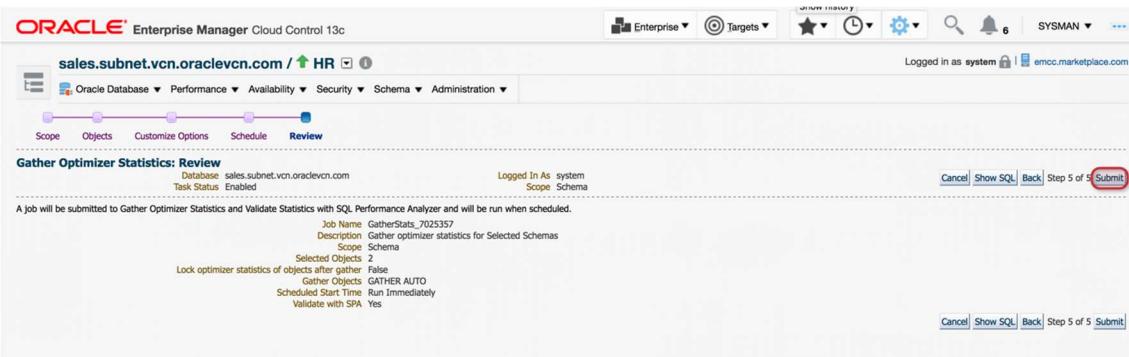
Click "OK"

**4.18 Click "Next"**

4.19 Click "Next"

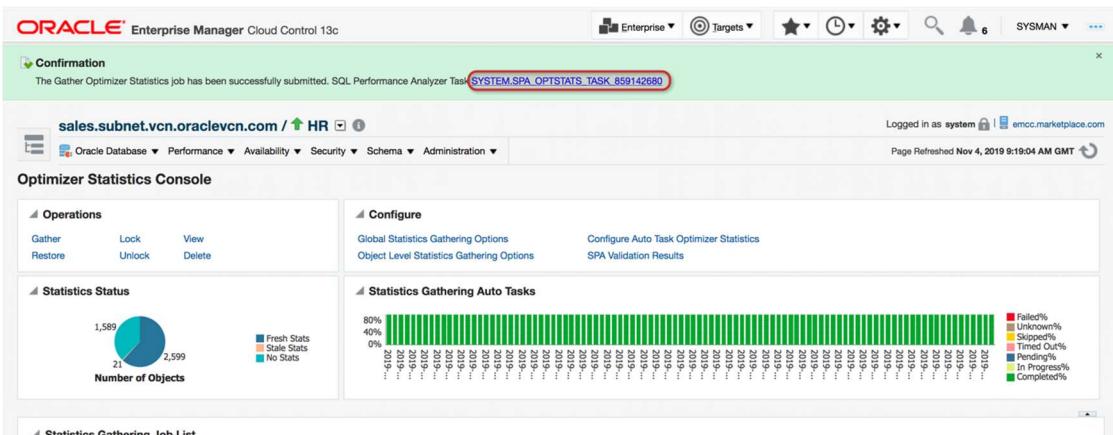


4.20 Click "Submit"



4.21 In the confirmation section on top, click on the 'SQL Performance Analyzer Task' that was started.

If you accidentally closed or lost this page, go to DB Target -> Performance Menu -> SQL Performance Analyzer Home -> Select the latest SPA task you just created at the bottom of the page



- 4.22** You have now a running SQL Performance Analyzer task. Wait until it “Last Run Status” becomes Completed.

Click on “Name”

Select	Name	Owner	Last Modified	Current Step Name	Type	Last Run Status	SQLs Processed	Steps Completed	Task Type	Description
<input checked="" type="radio"/>	SYSTEM.SPA_OPTSTATS_TASK_859142680	SYSTEM	Nov 4, 2019 9:20:16 AM	COMPARISON_IMPACT	Compare	Completed	25 of 25	7 of 7	Optimizer Stats	Validate pending stats with one-click SPA flow. Schemas: STAT1, STAT2

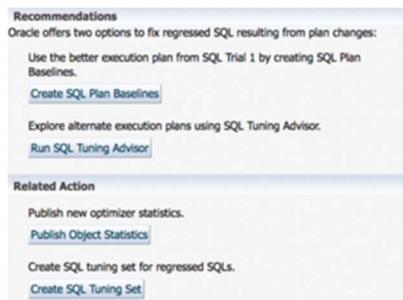
- 4.23** As you can see there have been four SQL trials executed. The first two have identified SQL statements with plan changes. In the last two trials it is only statements with plan changes that have been executed. This will reduce the amount of time and resources used in a production system.

Click on “” for the second report.

SQL Trial Name	Description	Created	SQL Executed	Status
FIRST_TRIAL	Trial with current optimizer statistics. Only explain execution plan.	11/4/19 9:19 AM	No	COMPLETED
SECOND_TRIAL	Trial with pending optimizer statistics. Only explain execution plan so to find SQLs with plan change.	11/4/19 9:19 AM	No	COMPLETED
THIRD_TRIAL	Trial with current optimizer statistics, only executing the SQLs with plan change.	11/4/19 9:19 AM	Yes	COMPLETED
FOURTH_TRIAL	Trial with pending optimizer statistics, only executing the SQLs with plan change.	11/4/19 9:19 AM	Yes	COMPLETED

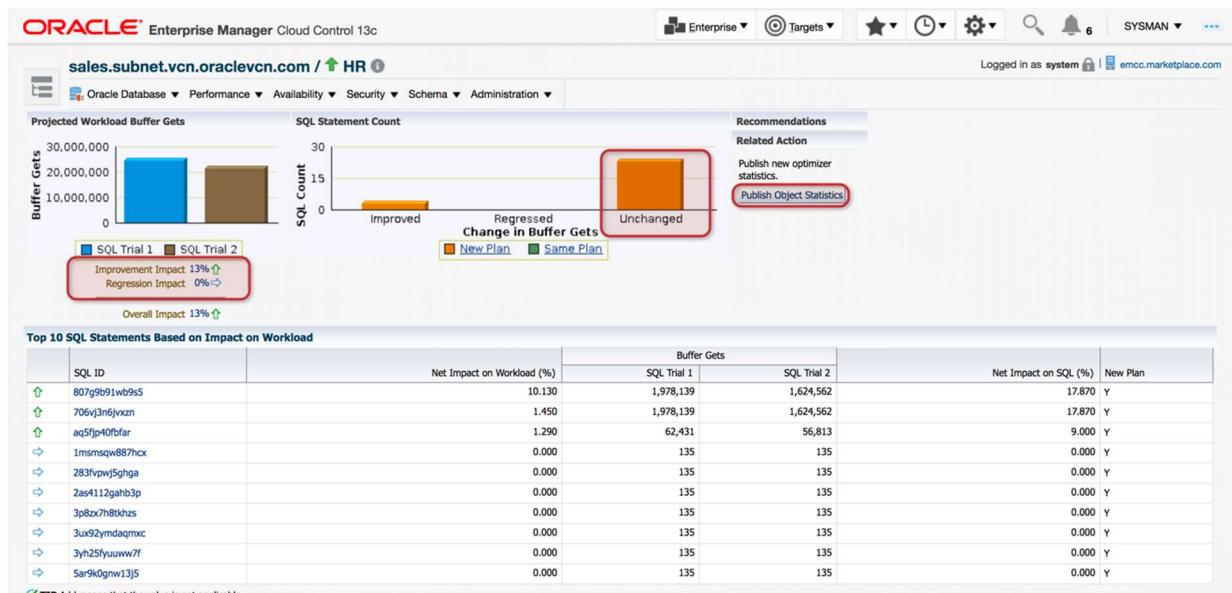
- 4.24** As we can see the majority of our statements had unchanged performance. We have a significant improvement but most important to notice is that we have no regression.

If there had been regression then we have the ability to tune the regressed statement or use SQL Plan Baselines to remediate the identified regressions. Note one can also use SQL Tuning Advisor to remediate regressions by implementing SQL Profile recommendations



Since this application has used stale statistics for a long period, then it would be good to have new statistics implemented.

- 4.25** Click on “Publish Object Statistics”



- 4.26** We can now change statistics for all tables where we have pending statistics. For the scope of this exercise we will only change statistics for schema STAT1.

Click the Checkbox for schema STAT1

Click "Publish"

Select	Schema	Count
<input type="radio"/>	MDSYS	1
<input type="radio"/>	DIJMSYS	2
<input checked="" type="radio"/>	STAT1	10
<input type="radio"/>	STAT2	12
<input type="radio"/>	SYS	105
<input type="radio"/>	SYSTEM	2

Click "Yes"

Confirmation

Are you sure you want to publish all the statistics for the schema STAT1?

No Yes

Information

Statistics of schema STAT1 has been published successfully.

For publishing statistics for user STAT2 please repeat the task in 4.25

You have now learned how to work with SPA. As you can see there are Guided Workflows that will help you during your analysis and verify that you can implement new changes in production with confidence.

Details about newly published statistics can be found if you go to 'Schema' -> 'Database Object' -> 'Tables' and select tables for schema 'STAT1'

E. Database Replay (Optional, time permitting)

Replay workloads against Pluggable Databases in the Container Database

Estimated Time to Complete Use Case: 10 minutes. Note this lab is optional, if you have completed the other recommended labs, you can go through this exercise. Database replay lab is a command line lab. There are two reasons for this.

1. We have seen that there is a majority of our customers preferring API and not Enterprise Manager.
2. There is currently a discrepancy between preprocessed version number and the version number for same database presented in Enterprise Manager. This have been fixed for 12.2 and older databases and the fix for 18.0 and beyond will be fixed in next monthly EM bundle patch

We have been asked to add 3 new indexes to the application, before these indexes can be added we need to proof that the performance in the database will be better. Since there are an additional cost to maintain indexes it is not enough to validate the performance with SQL Performance analyzer only. Replay will be done against the container database Sales. The changes need to be done in the OLTP container against the DWH_TEST schema. The database version is 18c so the capture is done against CDB and the replay will also be done against the CDB

5. Create a Replay Task

- 5.1 You need to connect two sessions to your dedicated host as user OPC using the provided SSH key.

Use putty or similar to connect to your local host. ([see section above: Lab Environment Setup](#))

- 5.2 `ssh -i [privatekey] opc@[Your IP]`

Session 1 and session 2

- 5.3 Connect to user oracle from the OPC user

`sudo su – oracle`

Session 1

- 5.4 Set Environment variables for sales database

`./sales.env`

- 5.5 Connect to sales database and create indexes. (indexes are already created we just need to make them visible)

```
sqlplus system/welcome@oltp
alter index dwh_test.DESIGN_DEPT_TAB2_IDX1 visible;
alter index dwh_test.DISTRIBUTION_DEPT_TAB2_IDX visible;
alter index dwh_test.OUTLETS_TAB3_IT_IDX visible;
exit
```

- 5.6 We have already performed the capture it is stored in /home/oracle/scripts/dbpack/RAT_CAPTURE/DBReplayWorkload_OLTP_CAP_1 RAT_REPLAY. The capture directory should be copied to a Replay directory. In a normal situations replay is performed against a test server. This test environment is limited so we will only copy the directory to a replay path instead

```
cd scripts/dbpack
cp -r RAT_CAPTURE/DBReplayWorkload_OLTP_CAP_1 RAT_REPLAY
cd RAT_REPLAY/DBReplayWorkload_OLTP_CAP_1
```

- 5.7 Connect to as sysdba and grant become user to system on all containers

```
sqlplus sys/welcome1 as sysdba
grant become user to system container=all;
```

- 5.8 Connect to system create a directory object to locate the capture and preprocess the capture

```
connect system/welcome1
CREATE DIRECTORY DBR_REPLAY AS
'/home/oracle/scripts/dbpack/RAT_REPLAY/DBReplayWorkload_OLTP_CAP_1';
exec DBMS_WORKLOAD_REPLAY.PROCESS_CAPTURE (capture_dir => 'DBR_REPLAY');
```

- 5.9 We can now start to replay the workload. Initialize replay will load replay metadata created during preprocessing

- 5.10 exec DBMS_WORKLOAD_REPLAY.INITIALIZE_REPLAY (replay_name => 'REPLAY_1', replay_dir => 'DBR_REPLAY');

- 5.11 If the replay environment uses different connect strings compared to the capture environment then we need to remap connections. Check connect strings.

```
select * from DBA_WORKLOAD_CONNECTION_MAP;
```

- 5.12 Remap connections

```
exec DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION (connection_id => 1, replay_connection => 'HR');
exec DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION (connection_id => 2, replay_connection => 'OLTP');
exec DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION (connection_id => 3, replay_connection => 'SALES');
exec DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION (connection_id => 4, replay_connection => 'SALES');
exec DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION (connection_id => 5, replay_connection => 'PSALES');
exec DBMS_WORKLOAD_REPLAY.REMAP_CONNECTION (connection_id => 6, replay_connection => 'SALES');
```

- 5.13 Check new settings for connect strings

```
select * from DBA_WORKLOAD_CONNECTION_MAP;
```

- 5.14 Prepare the replay by setting replay options. This replay will use default synchronization which is time-based synchronization. With this setting we will as good as possible honor timing for each individual call. If a session has slow SQL statements then other sessions will still honor its timing but they will not wait for the slow session. This can cause higher divergence. If divergence is less than 10 % then it should be considered as a good replay.

- 5.15 exec DBMS_WORKLOAD_REPLAY.PREPARE_REPLAY (synchronization => 'TIME');

```
*****
```

Now switch to session 2. You should already be connected as user oracle

```
*****
```

- 5.16 Set Environment variables for sales database and change to the replay directory
- 5.17 . ./sales.env
- 5.18 cd scripts/dbpack/RAT_REPLAY/DBReplayWorkload_OLTP_CAP_1
- 5.19 Calibrate the replay and validate how many replay clients that are needed to replay the workload.
- 5.20 Note! Replay clients are the application tier and should not be co-allocated with the database due to resource usage. Our recommendation is to place replay clients close to the database to avoid none wanted delays between database and replay clients. This is regardless if the application tier is located far away. The reason is that the replay clients communicate with the database to know when a certain database call should be replayed and if replay clients are located far away it will delay the call and create artificial delays during the replay.

```
wrc mode=calibrate replaydir=/home/oracle/scripts/dbpack/RAT_REPLAY/DBReplayWorkload_OLTP_CAP_1
```

```
*****
```

Calibrate output

```
*****
```

Workload Replay Client: Release 18.0.0.0.0 - Production on Tue Nov 5 09:43:45 2019

Copyright (c) 1982, 2018, Oracle and/or its affiliates. All rights reserved.

Report for Workload in:

/home/oracle/scripts/dbpack/RAT_REPLAY/DBReplayWorkload_OLTP_CAP_1

```
-----
```

Recommendation:

Consider using at least 1 clients divided among 1 CPU(s)

You will need at least 112 MB of memory per client process.

If your machine(s) cannot match that number, consider using more clients.

Workload Characteristics:

- max concurrency: 30 sessions
- total number of sessions: 534

Assumptions:

- 1 client process per 100 concurrent sessions
- 4 client processes per CPU
- 256 KB of memory cache per concurrent session
- think time scale = 100
- connect time scale = 100
- synchronization = TRUE

- 5.21 The workload is relatively small and it needs only one replay client so we will start it from this session

```
wrc system/welcome1@sales mode=replay  
replaydir=/home/oracle/scripts/dbpack/RAT_REPLAY/DBReplayWorkload_OLTP_CAP_1
```

```
*****
```

Session 1

```
*****
```

- 5.22 You should still be connected in the SQL*Plus session as used before. From this window start the replay

```
Exec DBMS_WORKLOAD_REPLAY.START_REPLAY();
```

- 5.23 Monitor the replay in session 2 and when the replay has finished the generate replay reports from session 1.

- 5.24 When replay has finished import capture AWR data. First create a common user as staging schema

```
Create user C##CAP_AWR;
grant DBA to C##CAP_AWR;
SELECT DBMS_WORKLOAD_CAPTURE.IMPORT_AWR (capture_id => 27,staging_schema => 'C##CAP_AWR') from dual;
```

- 5.25 Generate replay report as a text report. This report can also be generated in HTML or XML format.

```
Set long 500000
Set linesize 200
Set pagesize 0
Spool replay_report.txt
dbms_workload_replay.report (replay_id => 1, format=> 'TEXT') from dual;
spool off
```

- 5.26 Please open the text report with a Linux editor of your choice such as vi and look at replay details.

```
!vi replay_report.txt
```

- 5.27 Can you see if the replay uses more or less database time than the capture?

Exit the report in vi use "ZZ" and you will return to SQL*plus

- 5.28 Generate compare period report as HTML report.

```
spool compare_period_report.html
VAR v_clob CLOB
BEGIN dbms_workload_replay.compare_period_report(replay_id1 => 1, replay_id2 => null,
format => DBMS_WORKLOAD_REPLAY.TYPE_HTML, result => :v_clob);
END;
/
print v_clob;
spool off
exit
```

- 5.29 To be able to read the report it needs to be downloaded change file permissions and copy the file to /tmp

```
chmod 777 compare_period_report.html
```

```
cp compare_period_report.html /tmp
```

- 5.30 Use a scp client to copy the file to your local machine. Open the file in a text editor and remove initial lines before first row starting with

```
"<html lang="en">"
```

and trailing lines after last row ending with

```
"< b> End of Report. </b>
</body>
</html> "
```

You can now open the report in a browser and look at SQL statement with performance improvements and regression.

You have now finished the replay lab.

We have now seen how you can use Real Application Testing Database Replay to validate changes that may impact performance on both SQL statements and DML statements. We have also seen the extensive reporting that will help us to find and analyze bottlenecks or peaks during certain workloads. We hope that this have given you a good overview how to use the replay feature.

Appendix A. DB Performance Overview and Business Cases

Oracle Enterprise Manager Express

Oracle Database 12c introduced Oracle Enterprise Manager Express, or simply EM Express, a web based GUI tool optimized for performance Management. EM Express is extremely light weight and is built inside the database so it doesn't require any additional install. With only a 20 MB disk footprint, there is no resource usage when it is not invoked or used. You can use Oracle Enterprise Manager Express to manage a single database while Cloud Control 12c can be used to manage many databases and targets.

Oracle Enterprise Manager Database Express includes Performance Hub, a completely new unified interface for performance monitoring. It is the single pane of glass view of database performance with access to ADDM, SQL Tuning, Real-Time SQL Monitoring and ASH Analytics under the same hood. A flexible time picker allows the administrator to seamlessly switch between Real-Time and Historical views of database performance. For Oracle RAC databases, there is an additional RAC tab that allows the database administrator to monitor cluster related performance problems.

Diagnosing a slowly performing system or a sudden degradation in performance is a time consuming task and often the activity where the DBAs spend most of their time. A major component of the challenge is to simply identify what is causing the problem in our increasingly complex business environments. There is a need to sample the current state of all the sessions that are active in our systems, but analyzing this data for transient problems that occur for very short durations is not simple. EM Express allows the administrator to rollup, drilldown, and slice or dice performance data across various performance dimensions that are captured along with the session state. With the ability to create filters on various dimensions, identifying performance issues has never been easier.

Appendix B. Real Application Testing overview and Business Cases

A. SQL Performance Analyzer Optimizer Statistics

Business Case

When gathering new statistics, it is not uncommon that the new statistics cause the optimizer to choose a new query plan. In most cases the new plan will be more efficient but sometimes it causes query regression. As a DBA it is important to proactively predict how new statistics will change the overall performance in the database. With SQL Performance Analyzer (SPA) you have the ability to execute most of your SQL statements that occur in your database.

We have the ability to gather production SQL statements into SQL Tuning Sets either incrementally from Cursor Cache or AWR history for the period of time that we want to test. For the purpose of the lab, we have one pre-created SQL Tuning Set which is representative of the peak workload of this application. We have also gathered statistics in pending mode for validation. So, let's see if the new statistics will change the performance for this application.

Workflow overview

In this exercise we will learn to use the SPA guided workflow for Optimizer Statistics validation, how to tune regressed statements and how to implement pending statistics on tables.

This exercise will be done against a pluggable database, DW, in a container database 'test'.

The condition is that we have a warehouse based on 2 schemas 'STAT1' and 'STAT2'. The warehouse has been populated and used for a couple of month and we have currently not updated the statistics. We want to make sure that when we publish new statistics then we would not be surprised with a performance regression.

Some of the activities that are associated with this exercise are time consuming so we have performed those tasks in advance for lab purposes to save some time. The following task has already been executed in the setup of the environment:

- Gather Pending Optimizer Statistics on schema STAT1 and STAT2

Steps that we are going to perform in this exercise:

- Run SPA Optimizer Statistics workflow
- Compare performance of current statistics and pending statistics
- Use SQL Plan base lines to fix plan regressions
- Run SPA Optimizer Statistics workflow validate implemented SQL Plan Base lines
- Compare performance of current statistics and pending statistics + SQL Plan Baselines implemented
- Implement (Publish) new statistics

B. Database Replay

Business Case

During changes that can have impact on performance it is important to know that the server will handle all workloads. On this front we have the Database Replay feature that will help us to predict the workload behavior. We can test different scenarios to make sure that we are as much prepared as possible. Examples of scenarios to test could be:

- Worst case scenario during peaks.
- Growing user activity

In this use case we are going to replay a workload that have been pre-captured. For the purpose of the lab, we've already setup the test system and the Container database appropriately to the point-in-time of captures so that replay can be started.

Lab Overview

Objective:

The objective of this document is to provide high-level guidelines on new features associated Real Application Testing in Oracle Enterprise Manager Cloud Control 13c.

To perform real-world testing of Oracle databases, by capturing production workloads and replaying them on test systems enables you to perform real-world testing quickly and accurately. This allows enterprises to assess the impact of any planned system change before deploying it on production reliably. Additionally, with Oracle Database 12c and beyond we have the Multitenant capability that will make it possible to consolidate application into different pluggable databases within one container database. This allows us to reduce the footprint of each database and reduce resource usage on the server.