Oracle Cloud Infrastructure

Database Cloud Service Load Test

V2.0

ORACLE LAB BOOK | MAY 2018

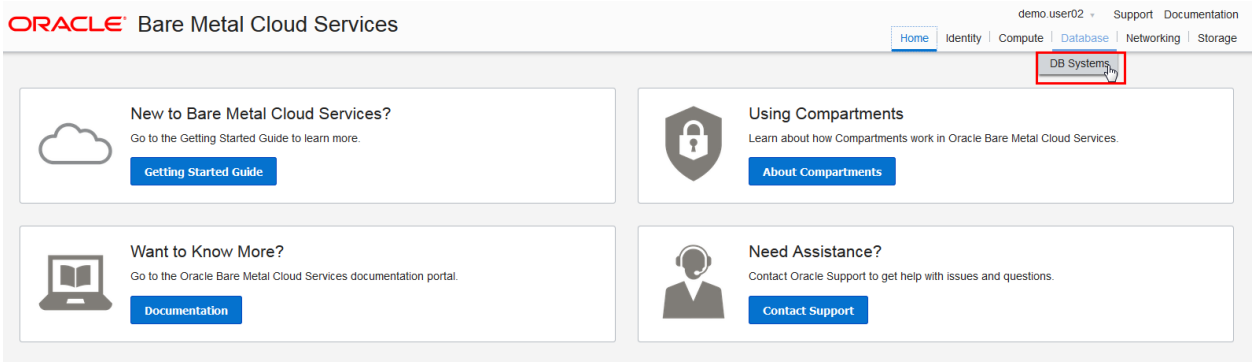
By
Oracle Sales Consulting Norway

1. Disclaimer

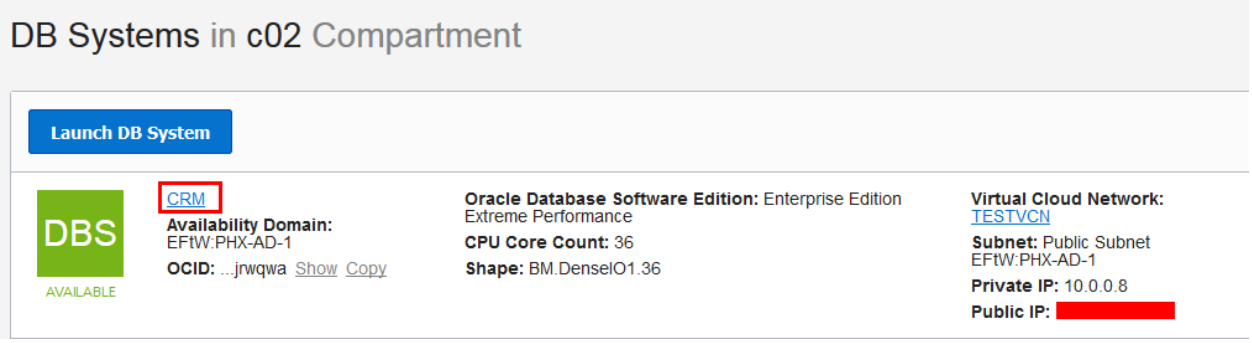
The following is intended to outline our general product direction. It is intended for information purposes Exercise 6: Configure and run Loadtest

1. Prepare Database for loadtesting:

At the Home page of Oracle Bare Metal Cloud Service. Lets select Database DB System to proceed to Database page.

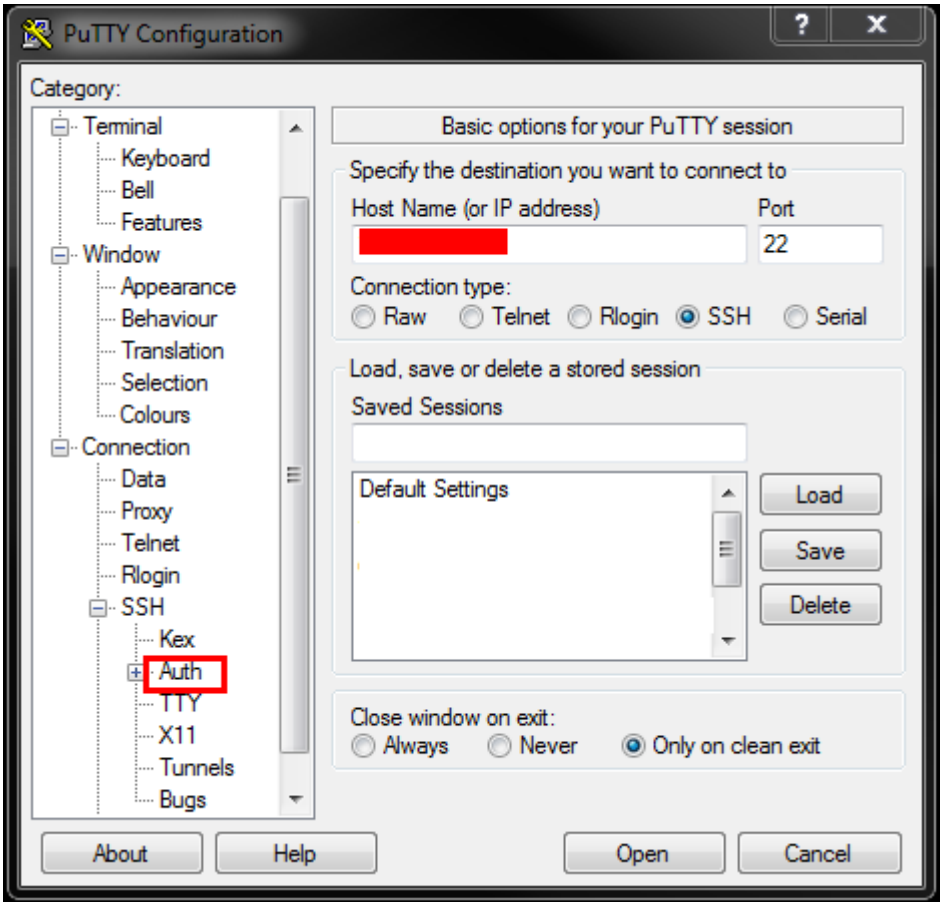


Once in Database page, you will see CRM DB System. If you don't see DB System, make sure you have selected the correct compartment.

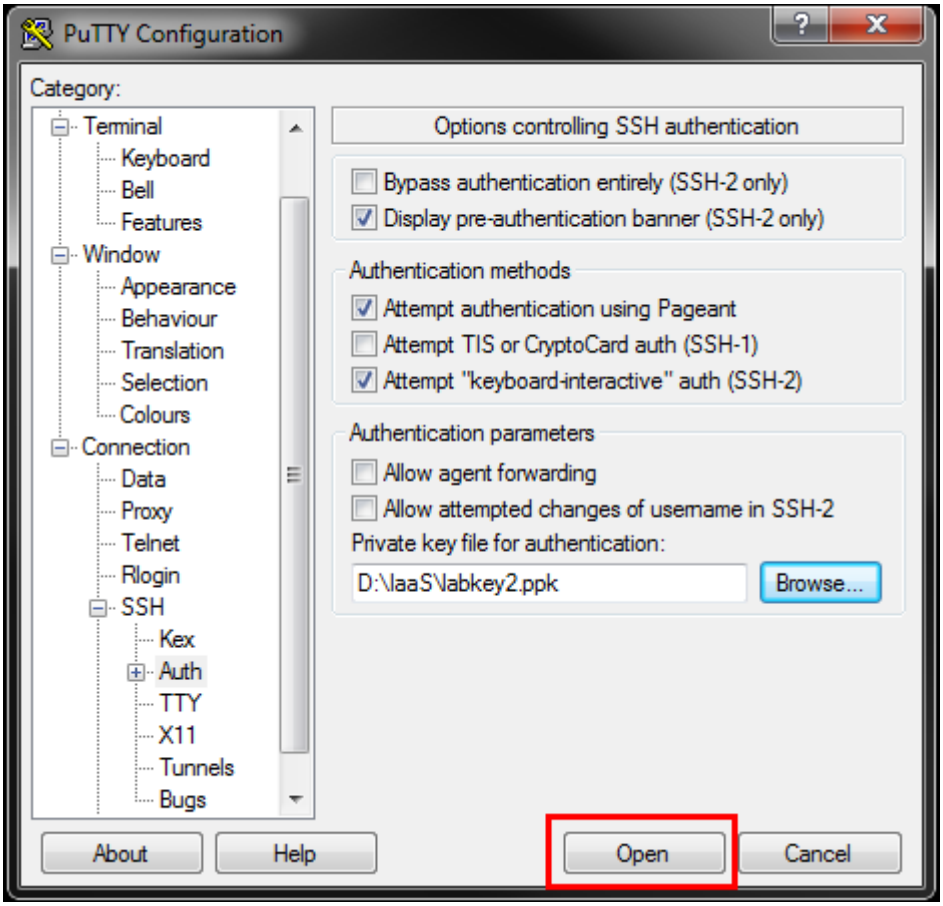


This page will give more details about the DB System like Availability Domain, Shape, Public IP etc. Note down the Public IP address.

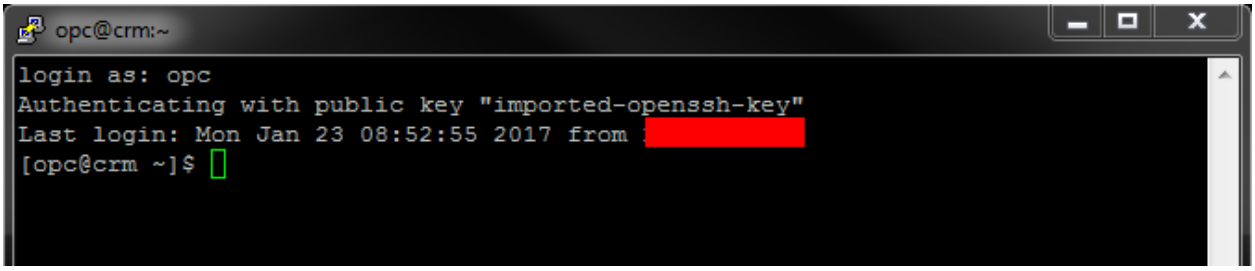
Launch Putty and enter the Public IP copied from above step



Browse and select the private key used to create the CRM system.



You will need to login as opc user to CRM instance.



Switch to root user

```
[opc@crm ~]$ sudo su -

[root@crm ~]#
```

Disable iptables firewall (as root)

```
[root@hr ~]# service iptables stop

iptables: Setting chains to policy ACCEPT: filter          [  OK  ]

iptables: Flushing firewall rules:                         [  OK  ]

iptables: Unloading modules:                               [  OK  ]

[root@hr ~]# chkconfig iptables off

[root@hr ~]#
```

Exit from root user and switch to Oracle User.

```
[root@crm ~]# exit

logout

[opc@crm ~]$ sudo su - oracle

[oracle@crm ~]$
```

Setup the environment by running the oraenv file

```
[oracle@crm ~]$ . oraenv

ORACLE_SID = [oracle] ? crm

The Oracle base has been set to /u01/app/oracle

[oracle@crm ~]$
```

Get the value for ORACLE_UNQNAME and set the environment variable – Note that this step is required for Transparent Data Encryption (TDE) to work. Our DBaaS instance enforces TDE.

```
[oracle@crm ~]$ srvctl config database

crm_phx372

[oracle@crm ~]$ export ORACLE_UNQNAME=crm_phx372

[oracle@crm ~]$
```

Use LSNRCTL to get the servicename for the PDB database. We will need this servicename later in the exercise

```
[oracle@crm ~]$ lsnrctl stat


LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 31-MAR-2017 10:04:22


Copyright (c) 1991, 2014, Oracle.  All rights reserved.


Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                30-MAR-2017 12:24:29
Uptime                    0 days 21 hr. 39 min. 53 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/12.1.0.2/grid/network/admin/listener.ora
Listener Log File         /u01/app/grid/diag/tnslsnr/hr/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.0.2.2)(PORT=1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=hr.sub03240756572.nolabc14.oraclevcn.co
m)(PORT=5500))(Security=(my_wallet_directory=/u01/app/oracle/admin/crm_phx372/xdb_
wallet))(Presentation=HTTP)(Session=RAW))
```

```
Services Summary...

Service "+ASM" has 1 instance(s).

  Instance "+ASM1", status READY, has 1 handler(s) for this service...

Service "crmXDB.sub03240756572.nolabc14.oraclevcn.com" has 1 instance(s).

  Instance "crm", status READY, has 1 handler(s) for this service...

Service "crm_phx372.sub03240756572.nolabc14.oraclevcn.com" has 1 instance(s).

  Instance "crm", status READY, has 1 handler(s) for this service...

Service "pdb1.sub03240756572.nolabc14.oraclevcn.com" has 1 instance(s).

  Instance "crm", status READY, has 1 handler(s) for this service...

The command completed successfully

[oracle@crm ~]$
```

Login to sqlplus and run the following commands to prepare for loadtesting and enable Enterprise Manager Express

alter system set cpu_count=36 scope=spfile;

alter system set processes=2000 scope=spfile;

exec DBMS_XDB_CONFIG.SETHTTPSPORT(5500);

```
[oracle@crm ~]$  sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 31 07:47:50 2017

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Connected to:

Oracle Database 12c EE Extreme Perf Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options

SQL> alter system set cpu_count=36 scope=spfile;
```

```
System altered.

SQL> alter system set processes=2000 scope=spfile;

System altered.

SQL> exec DBMS_XDB_CONFIG.SETHTTPSPORT(5500);

PL/SQL procedure successfully completed.

SQL>
```

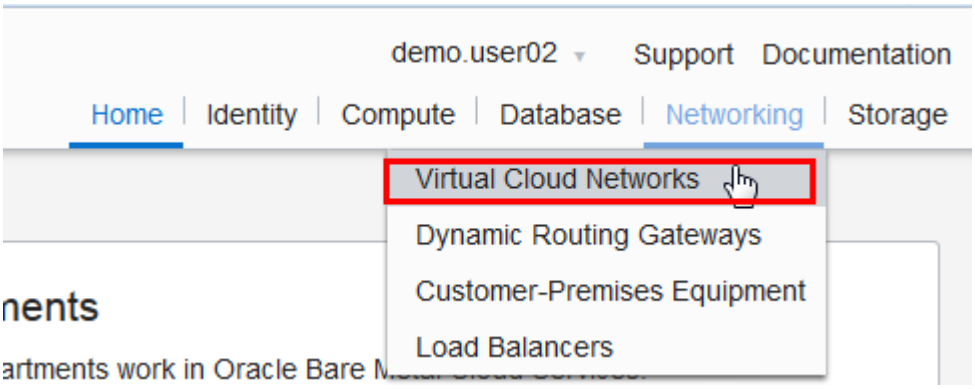
Shutdown the database and disable Archiving in order to save space during the loadtesting.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.
Total System Global Area 8589934592 bytes
Fixed Size                  2944952 bytes
Variable Size            2063597640 bytes
Database Buffers         4764729344 bytes
Redo Buffers               30609408 bytes
In-Memory Area           1728053248 bytes
Database mounted.
SQL> alter database noarchivelog;
Database altered.
SQL> alter database open;
Database altered.
SQL>
```

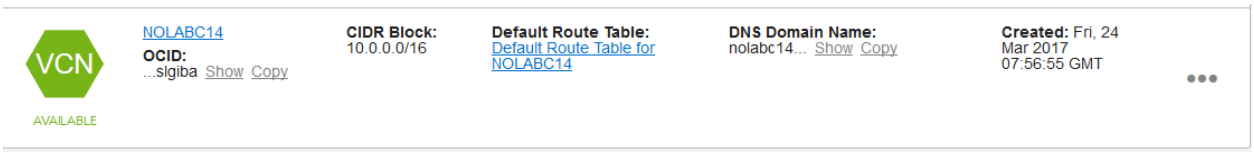
Exit from Sql*Plus

2. Edit the Virtual Network Setting to allow you to run Enterprise Manager Express

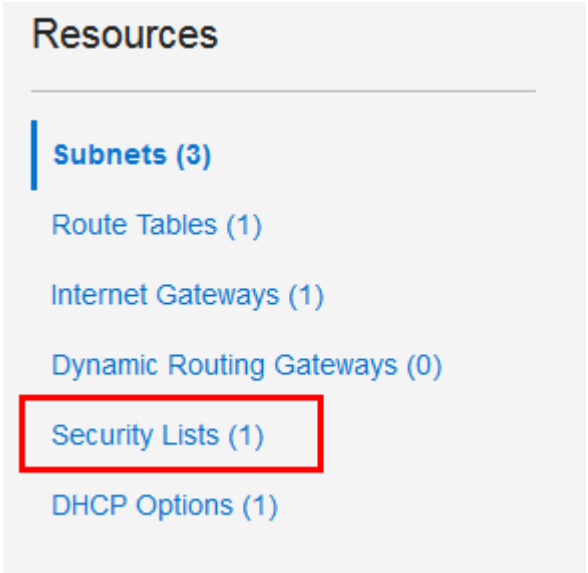
Go to Oracle Bare Metal Cloud Service console and select Virtual Cloud Networks under Networking.



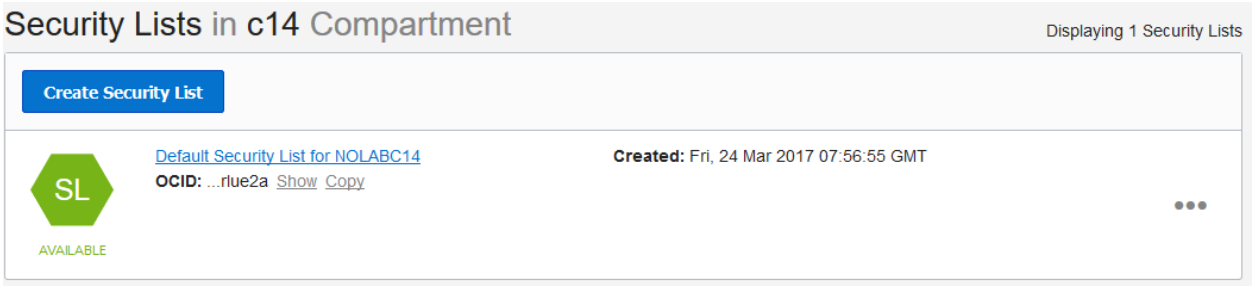
Click on the NOLABxx network you created earlier



On the left hand side, you can see Security List under Resources.

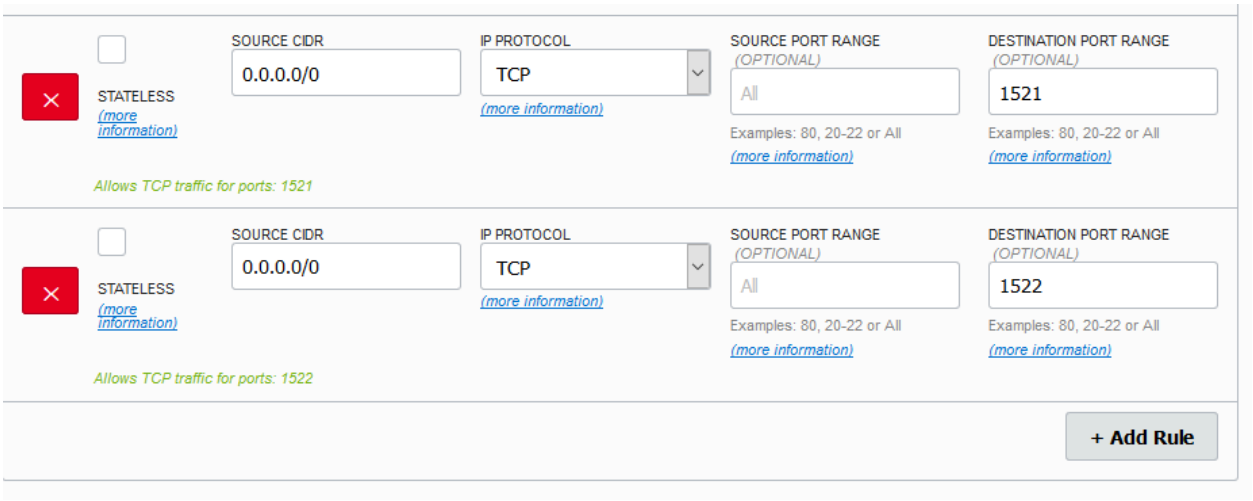


Click on the Security List, and you will see the Security Lists defined for your Virtual Network



Click on Default Security List for NOLABxx

Click on Edit All Rules



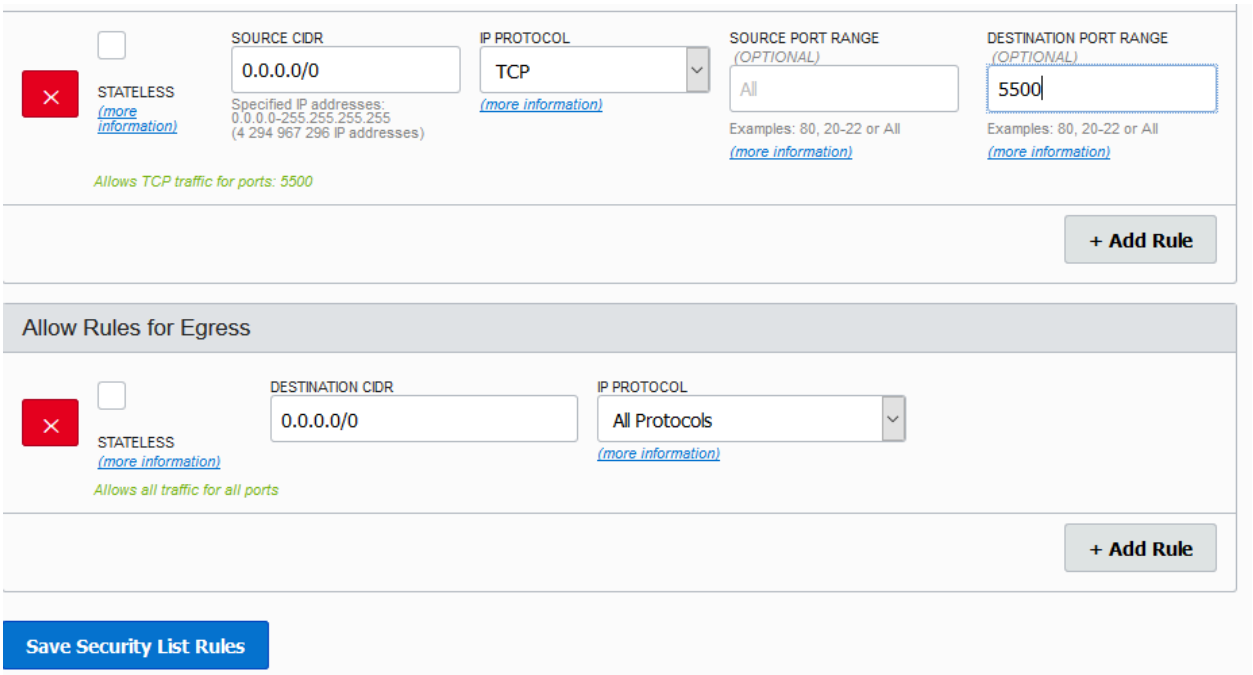
Click on Add Rule. You will now add a new Ingress rule for Enterprise Manager on port 5500

Add the entries like below and click Save Security List Rules

SOURCE CIDR: 0.0.0.0/0

IP PROTOCOL: TCP

DESTINATION PORT RANGE: 5500



3. Launch Database Express 12c

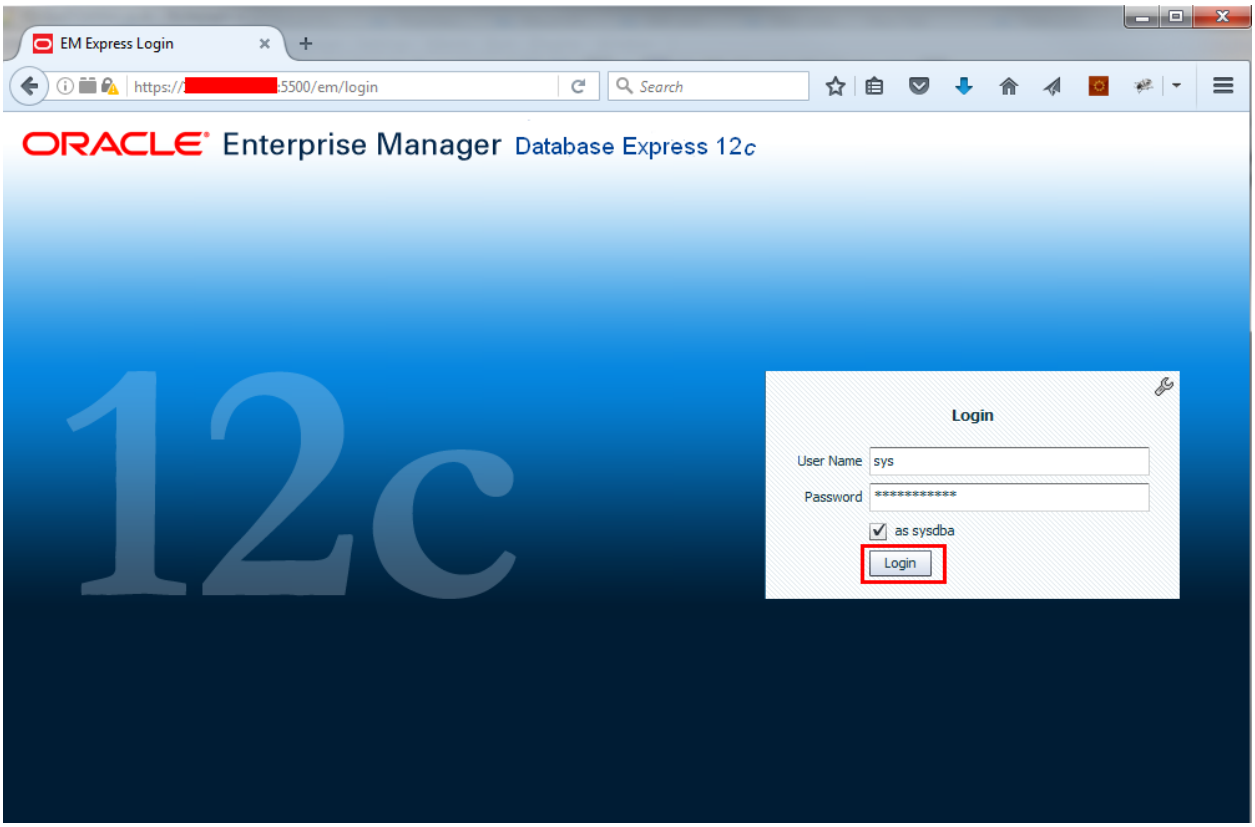
From browser Launch Database Express 12c with URL : https://<ip-address>:5500/em

Confirm security exception

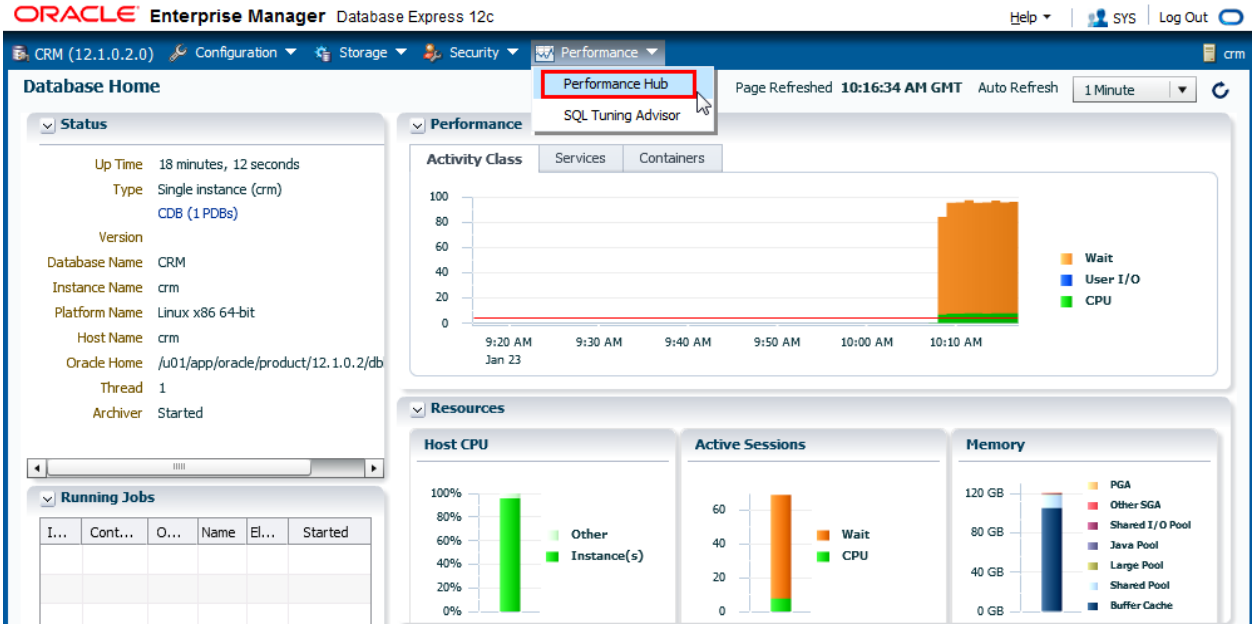
username: sys

password: sys password used when creating the DBaaS instance

Check “as sysdba”



Navigate to Performance Hub under Performance tab



4. Setup the client to run the loadtest tool.

Launch Putty and enter the Public IP for the VM where you installed the Oracle client in Exercise 5

```
Using username "opc".

Authenticating with public key "rsa-key-20170105"

[opc@labvm01 ~]$ sudo su - oracle

[oracle@labvm01 ~]$
```

Create a directory for the loadtest tool and use wget to download the tool. Then unzip the tool.

```
 [oracle@labvm01 ~]$ mkdir loadtest

[oracle@labvm01 ~]$ cd loadtest

[oracle@labvm01 loadtest]$ wget
http://129.146.24.100/downloads/load_test_package.zip

--2017-03-31 08:34:08--  http://129.146.24.100/downloads/load_test_package.zip

Connecting to 129.146.24.100:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK

Length: 6584 (6.4K) [application/zip]

Saving to: “load_test_package.zip”


100%[======================================>] 6,584       --.-K/s   in 0s


2017-03-31 08:34:08 (688 MB/s) - “load_test_package.zip” saved [6584/6584]


[oracle@labvm01 loadtest]$ unzip load_test_package.zip

Archive:  load_test_package.zip

  inflating: init_load.sh

  inflating: load_env.sh

   creating: scripts/

  inflating: scripts/init_check.sh

  inflating: scripts/init_object.sh

  inflating: scripts/init_user.sh

  inflating: scripts/load_object.sh

  inflating: scripts/load_object_network.sh

  inflating: scripts/stop_check.sh

  inflating: start_load.sh

  inflating: stop_load.sh

   creating: tmp/

  inflating: variable_load.sh

[oracle@labvm01 loadtest]$
```

Log on to the PDB using the servicename collected earlier in the exercise, and create the tablespace needed for the loadtest.

```
[oracle@labvm01 loadtest]$ sqlplus
sys/MAnager_2017#@//129.146.24.100/pdb1.sub03240756572.nolabc14.oraclevcn.com as
sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 31 10:16:15 2017

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Connected to:

Oracle Database 12c EE Extreme Perf Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options

SQL> SELECT name, open_mode, restricted FROM v$pdbs;

NAME                           OPEN_MODE  RES

------------------------------ ---------- ---

PDB1                           READ WRITE NO

SQL> create bigfile tablespace demo datafile size 200g autoextend on;

Tablespace created.

SQL>
```

5. Run the loadtest

The load uses a single schema and creates a test table for each session that will be connected, so the load needs to be initialized based on the maximum number of sessions expected for testing.

Setup and Execution

- Edit the load_env.sh file to setup variables for your test. This is where you will define the maximum number of test tables you will need. Edit the load_env.sh file to match the output below. Make sure to use your IP-addr and servicename.

```
[oracle@labvm01 loadtest]$ more load_env.sh

#!/bin/bash
```

```
###

### Variables

###

export TABLESPACE=demo

export USER_NAME=loadtest

export PW=WelcomE_2017#

export CONNECT_STRING=//129.146.24.100/pdb1.sub03240756570.nolabc14.oraclevcn.com

### test table name prefix

export TEST_TABLE=large_block

export MAX_TABLES=50

export OUT_LOG=load.log

[oracle@labvm01 loadtest]$
```

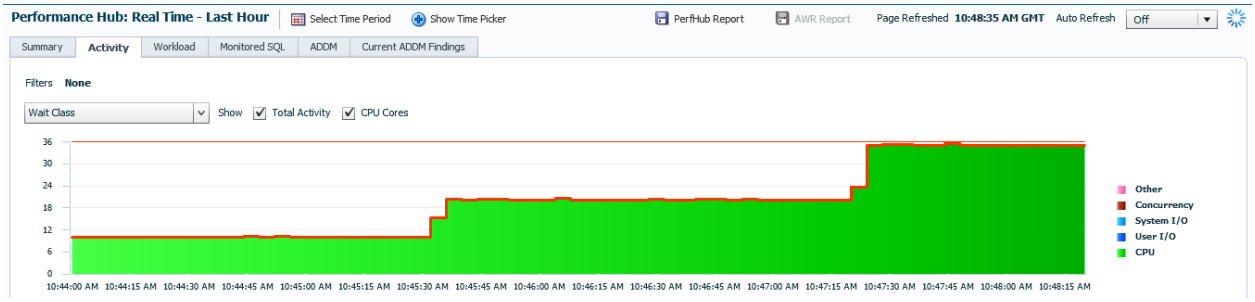
- Run the init_load.sh script to setup your user test schema and test tables. You will be prompted for the SYSTEM user password.

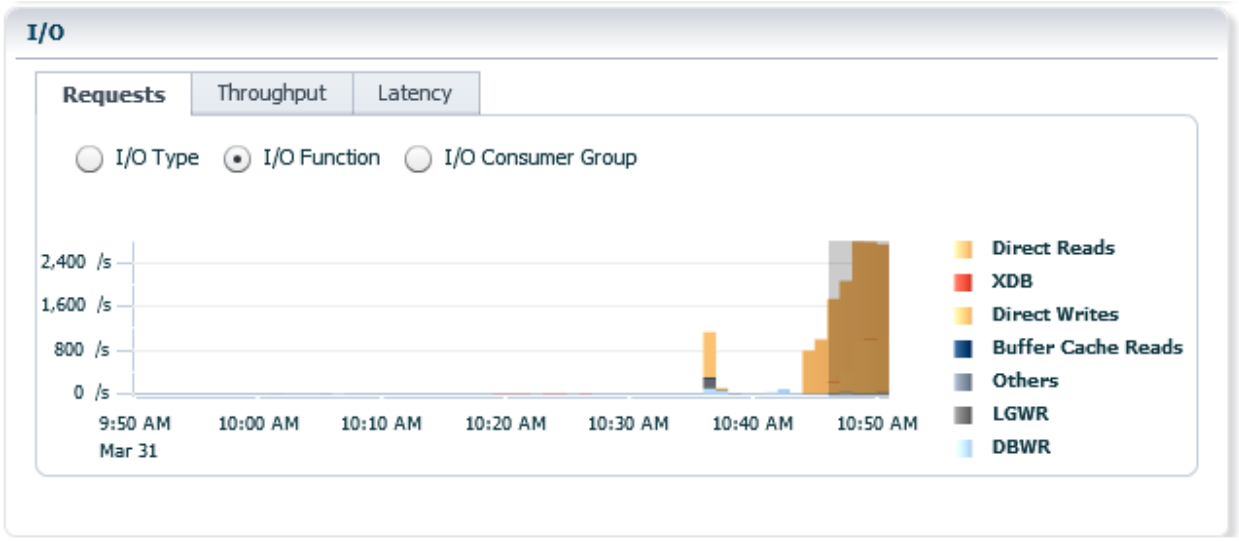
```
[oracle@labvm01 loadtest]$ . init_load.sh

System Password:

PL/SQL procedure successfully completed.

Objects still being initialized... 50

Objects still being initialized... 50

Objects still being initialized... 50

Objects still being initialized... 50

Objects still being initialized... 50



[oracle@labvm01 loadtest]$
```

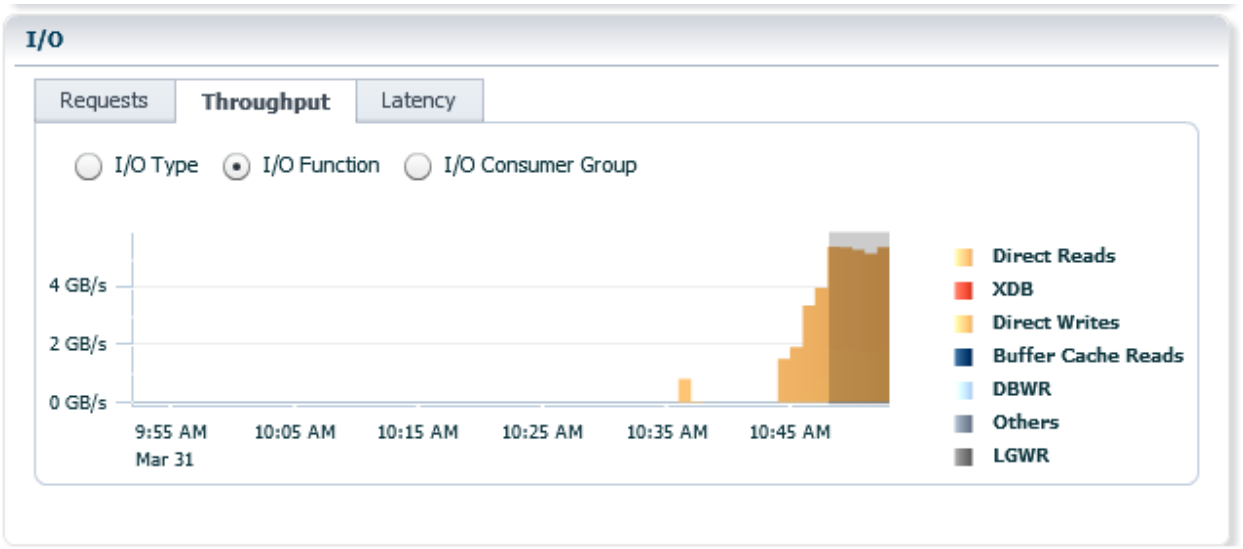
- Run the start_load.sh script to begin your test. This script requires two parameters, the low and high values for the test tables to use and thus the number of sessions. In this test we will ramp up the test in three steps. Use Enterprise Manager to monitor the performance after each load has been started.

```
[oracle@labvm01 loadtest]$ . start_load.sh 1 10

[oracle@labvm01 loadtest]$
```

When looking at the performance in Enterprise Manager you will se that the system will deliver more than 2500 IO request/sec and a throughput of over 5 GB/sec, without any IO wait in the database.







Use the stop_load.sh script to kill all sessions and stop the load.

```
[oracle@labvm01 loadtest]$ . stop_load.sh
```