

Comparison of Cloud Databases 2019Spring

Cloud computing 5525 0001



MAY 10

Grace Stalin (16291038)
Abdulsalam Elsaidi (12342281)



Abstract

There are a lot of cloud services in the market right now trending and the companies are finding it difficult to choose from their numerous options to make it easy, we have taken two different type of databases from different cloud combined them with another virtual machine using the third party tool called solar winds to track their performance.

Different types of database are tracked like one is made from scratch in VMs and the other from cloud database service.

In market there are currently using high pitch performance evaluation in which we are interested to implement the real time scenario, hence, we choose solar winds instead of a database benchmarking like sysbench, swing bench, blob etc.

We will furnish how we achieved that in the following document; the fortifying different ways of databases creation and their service usage how it will affect the performance rate will be clearly captured in this project.

Acknowledgement

First of all, we would like to thank The Almighty Lord who made this possible.

We would like to say our sincere gratitude to the cloud computing prof.Baek Young Choi who is the guidance and inspiration of our project achievement.

Every successful individual knows that his/her achievement depends on their cooperating family who lifts you when you are down hence heartfelt thanks to our families.

To our rivals who inspired us to be resilient with perseverance and to our friends who helped us with motivation and ideas.

Table of contents

1.Introduction.....	5
a)project motivation	
b)objectives and aims	
2.Description of the architectural image.....	7
a)parameters	
b)Languages	
c)Platforms	
d)Tools	
3.Process1.....	9
a)EC2 prerequisite	
b)Challenges and rectification	
c)Finally database in the EC2	
4.Process2.....	16
a)Oracle database service	
b)Challenges	
c)changes with DB mounting	
5.Process3.....	20
a)GCP VM creation	
b)prerequisite for performance evaluation tool	
c)Connecting the clouds	
6.process4.....	31
a)Iterations	
b)Initial performance	
c)Final analysis	
d)Conclusion	

Introduction

Project Motivation:

This cloud database service field has been inspired by the interest we have on the “cloud computing” and their resources available for students to explore and learn however we wanted to be unique from other projects which explains cloud backend Work.

In the beginning, we don't have much knowledge about the project and difficult to choose then through some guidance we started to explore and did this with so many trials and errors with last minute changes; etc.

Our effort should not be a replica of the previous project but it should be useful for the real time usage and if we include this in our resume definitely it's a big add on to the achievements.

Hence our project has enable us to give the real time analyzing experience and evaluation criteria to develop in depth knowledge as startup learners with zero experience in the field.

Objective and Aims:

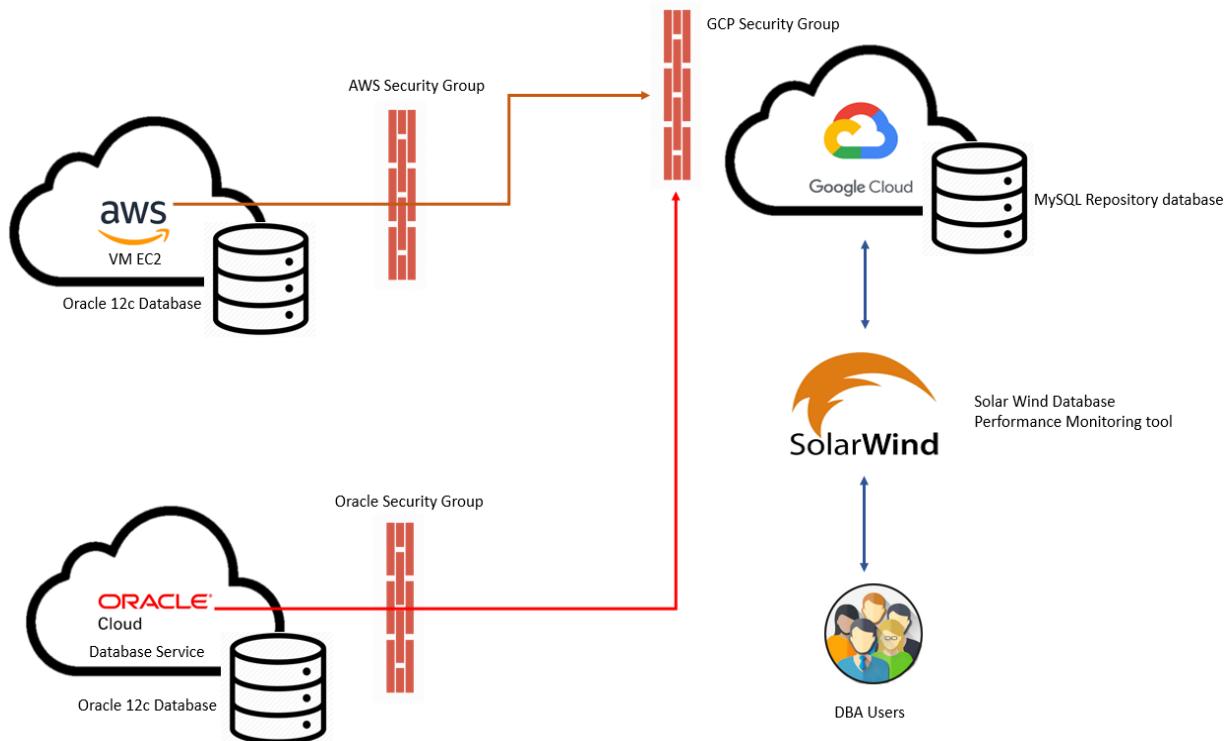
Identify, understand, and describe the performance evaluation tools in market. Services that the cloud will provide and how they work to meet the necessity of the corporates.

Server setup should be learned by using the reference videos and doing research on what type of performance analysis is in trending the market and type of competitors on them.

Minimizing the cost, as we students, cannot afford the cost of the cloud service providers hence keeping the cost effective in mind and doing the base work for that.

Parameters for analyzing the performance which will impact what type of changes in the database behavior and evaluating references and graphs representation should be concentrated for the better understanding.

Description of the architectural image:



This image will give the clarity of overall structure of all the processes that we are going to implement to achieve what the project main purpose.
Firewalls should be handled in each and every cloud to be connected with.
Handling them using the subnets channeling them in a public way and connection is done.
Hence process flow is clearly visualized.

Parameters, Tools, and Platforms:

Parameters which will impact the database performance are write, read, and their wait time to run the threads.

Commit and transactions are taken by the database.

Tools:

The performance evaluation tools used here are solar winds the main trending tool in current markets and their equal competition are IDERA.

Platforms:

Windows 2012RS

Linux(Rhel 7.2)

SQL cloud developer

Oracle DB

My SQL

Unix commands.

Process1: AWS EC2 Prerequisite step by step process:

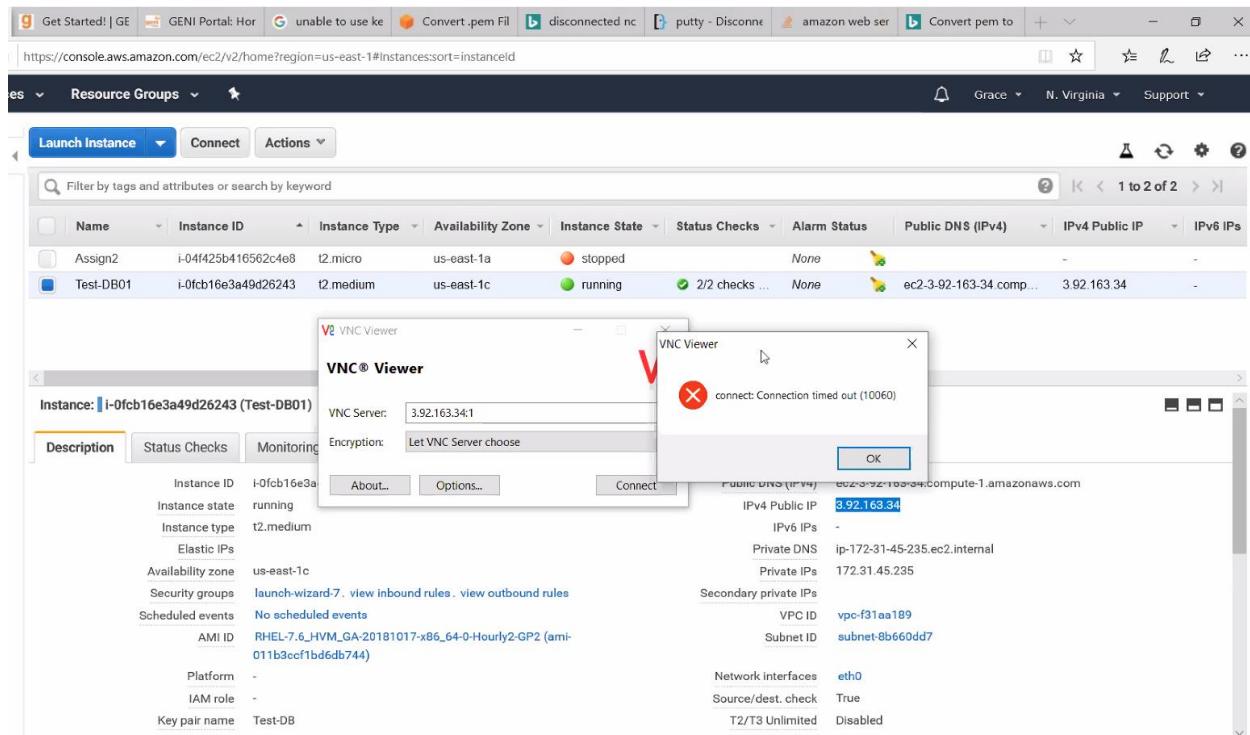
Step1: To Achieve our performance evaluation as a first step we need to create cloud accounts in private clouds like AWS, Oracle Cloud, and Google Cloud Platform and activate your educative version or free tier account.

Step2: As of the second step in our database project, we prerequisite the AWS settings for oracle DB to setup in AWS.

The reason why we have chosen oracle DB instead of other Databases like SQL is if we have to perform the SQL, we need windows OS which will cause more billing in private cloud platforms in order to reduce the cost and make it cost effective, we are setting a Linux machine with EBS volume of 100GB.

The prerequisite set up for AWS Linux server requires RPM (redhat package manager) which is a free and open source manager for Linux redhat.

Step3: The installation screen shots of those prerequisites rpm manager package using putty and SSH keys.



```

[root@ip-172-31-45-235 ~]#
[root@ip-172-31-45-235 ~]# yum -y install xterm xorg-x11* tigervnc-server*
Loaded plugins: amazon-id, rhui-lb, search-disabled-repos
rhui-REGION-client-config-server-7
rhui-REGION-rhel-server-releases
rhui-REGION-rhel-server-rh-common
(1/7)  [rhui-REGION-client-config-server-7-1.0.6-6.el7.noarch]

```

Pre-request installation

Installation of required rpm package from oracle repository.

```

[root@ip-172-31-45-235 ~]# yum install oracle-rdbms-server-11gR2-preinstall
Loaded plugins: amazon-id, rhui-lb, search-disabled-repos
o17_UERKR5
o17_latest
(1/5): o17_UERKR5/x86_64/updateinfo | 1.2 kB 00:00:00
(2/5): o17_UERKR5/x86_64/primary | 1.4 kB 00:00:00
(3/5): o17_latest/x86_64/group | 31 kB 00:00:00
(4/5): o17_latest/x86_64/updateinfo | 3.0 MB 00:00:00
(5/5): o17_latest/x86_64/primary | 810 kB 00:00:00
o17_UERKR5
o17_latest

```

```
[root@ip-172-31-45-235 ~]# yum repolist
[root@ip-172-31-45-235 yum.repos.d]# cd
[root@ip-172-31-45-235 ~]# yum install oracle-rdbms-server-11gR2-preinstall
Loaded plugins: amazon-id, rhui-lb, search-disabled-repos
ol7_UEKR5
ol7_latest
(1/5): ol7_UEKR5/x86_64/updateinfo
(2/5): ol7_UEKR5/x86_64/primary
(3/5): ol7_latest/x86_64/group
(4/5): ol7_latest/x86_64/updateinfo
(5/5): ol7_latest/x86_64/primary
ol7_UEKR5
ol7_latest
Resolving Dependencies
--> Running transaction check
---> Package oracle-rdbms-server-11gR2-preinstall.x86_64 0:1.0-6.el7 will be installed
---> Processing Dependency: bc for package: oracle-rdbms-server-11gR2-preinstall-1.0-6.el7.x86_64
---> Processing Dependency: gcc for package: oracle-rdbms-server-11gR2-preinstall-1.0-6.el7.x86_64
---> Processing Dependency: sysstat for package: oracle-rdbms-server-11gR2-preinstall-1.0-6.el7.x8
```

```
Installed:
  oracle-rdbms-server-11gR2-preinstall.x86_64 0:1.0-6.el7

Dependency Installed:
  bc.x86_64 0:1.06.95-13.el7          bind-libs.x86_64 32:9.9.4-73.el7_6      bind-utils.x86_64 32:9
  compat-libstdc++-33.x86_64 0:3.2.3-72.el7  cpp.x86_64 0:4.8.5-36.0.1.el7_6.1    gcc.x86_64 0:4.8.5-36.
  glibc-devel.x86_64 0:2.17-260.0.15.el7_6.3  glibc-headers.x86_64 0:2.17-260.0.15.el7_6.3  gssproxy.x86_64 0:0.7.
  kernel-headers.x86_64 0:3.10.0-957.10.1.el7  keyutils.x86_64 0:1.5.8-3.el7        ksh.x86_64 0:20120801-
  libaio-devel.x86_64 0:0.3.109-13.el7       libbasicobjects.x86_64 0:0.1.1-32.el7   libcollection.x86_64 0
  libini_config.x86_64 0:1.3.1-32.el7       libmpc.x86_64 0:1.0.1-3.el7        libnfsidmap.x86_64 0:0
  libref_array.x86_64 0:0.1.5-32.el7       libstdc++-devel.x86_64 0:4.8.5-36.0.1.el7_6.1  libtirpc.x86_64 0:0.2.
  lm_sensors-libs.x86_64 0:3.4.0-6.20160601gitf9185e5.el7  mailx.x86_64 0:12.6-19.el7    mpfr.x86_64 0:3.1.1-4.
  psmisc.x86_64 0:22.20-15.el7           quota.x86_64 1:4.01-17.el7        quota-nls.noarch 1:4.0
  smartmontools.x86_64 1:6.5-1.el7         sysstat.x86_64 0:10.1.5-17.el7   tcp_wrappers.x86_64 0:

Dependency Updated:
  bind-libs-lite.x86_64 32:9.9.4-73.el7_6      bind-license.noarch 32:9.9.4-73.el7_6      glibc.x86_64 0:2.17-260.0.15.el7_6.3
  libgcc.x86_64 0:4.8.5-36.0.1.el7_6.1        libgomp.x86_64 0:4.8.5-36.0.1.el7_6.1    libstdc++.x86_64 0:4.8.5-36.0.1.el7_6.3

Complete!
[root@ip-172-31-45-235 ~]#
```

Step4: Create a user and groups to run oracle database on oracle user.

```
[root@ip-172-31-45-235 ~]#
[root@ip-172-31-45-235 ~]#
[root@ip-172-31-45-235 ~]# groupadd -g 54321 oinstall
groupadd: group 'oinstall' already exists
[root@ip-172-31-45-235 ~]# groupmod -g 54321 oinstall
[root@ip-172-31-45-235 ~]# groupmod -g 54322 dba
[root@ip-172-31-45-235 ~]# groupmod -g 54323 oper
groupmod: group 'oper' does not exist
[root@ip-172-31-45-235 ~]# groupadd -g 54323 oper
[root@ip-172-31-45-235 ~]# useradd -g oinstall -G dba,oper oracle
useradd: user 'oracle' already exists
[root@ip-172-31-45-235 ~]# usermod -g oinstall -G dba,oper oracle
[root@ip-172-31-45-235 ~]#
```

Step5: Set password for oracle user.

```
[root@ip-172-31-45-235 ~]#
[root@ip-172-31-45-235 ~]#
[root@ip-172-31-45-235 ~]# passwd oracle
Changing password for user oracle.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-45-235 ~]#
```

Step6:Disable Selinux permission in order to take privileges for redhat linux in AWS.

```
[root@ip-172-31-45-235 ~]#
[root@ip-172-31-45-235 ~]# passwd oracle
Changing password for user oracle.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-45-235 ~]# vi /etc/selinux/config
[root@ip-172-31-45-235 ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted

[root@ip-172-31-45-235 ~]#
```

Step7: Stop and disable the Linux firewall services.

```
[root@ip-172-31-45-235 ~]#
[root@ip-172-31-45-235 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
  Active: inactive (dead)
    Docs: man:firewalld(1)
[root@ip-172-31-45-235 ~]# systemctl stop firewalld
[root@ip-172-31-45-235 ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@ip-172-31-45-235 ~]#
```

Step8: Create new directory for oracle binary installation.

```
[root@ip-172-31-45-235 ~]#  
[root@ip-172-31-45-235 ~]#  
[root@ip-172-31-45-235 ~]# mkdir -p /u01/app/oracle/product/11.2.0.4/db_1  
[root@ip-172-31-45-235 ~]# chown -R oracle:oinstall /u01  
[root@ip-172-31-45-235 ~]# chmod -R 775 /u01  
[root@ip-172-31-45-235 ~]#
```

Step9:Download oracle database software from oracle e-delivery.

```
[oracle@ip-172-31-45-235 software]$ ll  
total 2295592  
-rw-rw-r-- 1 oracle oinstall 1239269270 Mar 23 12:17 linux.x64_11gR2_database_1of2.zip  
-rw-rw-r-- 1 oracle oinstall 1111416131 Mar 23 12:20 linux.x64_11gR2_database_2of2.zip  
[oracle@ip-172-31-45-235 software]$
```

Step10:Unzip the oracle database package

```
inflating: database/stage/Components/oracle.sysman.console.db/11.2.0.1.0/  
inflating: database/stage/Components/oracle.sysman.console.db/11.2.0.1.0/  
[oracle@ip-172-31-45-235 software]$ unzip linux.x64_11gR2_database_1of2.zip  
[oracle@ip-172-31-45-235 software]$ unzip linux.x64_11gR2_database_2of2.zip  
[oracle@ip-172-31-45-235 software]$
```

Step11:Installation of database package.

Go to the software location.

```
database  linux.x64_11gR2_database_1of2.zip  linux.x64_11gR2_database_2of2.zip  
[oracle@ip-172-31-45-235 software]$ ll  
total 2295592  
drwxr-xr-x 8 oracle oinstall          128 Aug 20  2009 database  
-rw-rw-r-- 1 oracle oinstall 1239269270 Mar 23 12:17 linux.x64_11gR2_database_1of2.zip  
-rw-rw-r-- 1 oracle oinstall 1111416131 Mar 23 12:20 linux.x64_11gR2_database_2of2.zip  
[oracle@ip-172-31-45-235 software]$
```

```

total 2295592
drwxr-xr-x  8 oracle oinstall          128 Aug 20  2009 database
-rw-rw-r--  1 oracle oinstall 1239269270 Mar 23 12:17 linux.x64_11gR2_database_
-rw-rw-r--  1 oracle oinstall 1111416131 Mar 23 12:20 linux.x64_11gR2_database_
[oracle@ip-172-31-45-235 software]$ cd database/
[oracle@ip-172-31-45-235 database]$ ls
doc  install  response  rpm  runInstaller  sshsetup  stage  welcome.html
[oracle@ip-172-31-45-235 database]$ ll
total 16
drwxr-xr-x 12 oracle oinstall  203 Aug 17  2009 doc
drwxr-xr-x  4 oracle oinstall  223 Aug 15  2009 install
drwxrwxr-x  2 oracle oinstall   61 Aug 15  2009 response
drwxr-xr-x  2 oracle oinstall   34 Aug 15  2009 rpm
-rw-rx-r-x  1 oracle oinstall 3226 Aug 15  2009 runInstaller
drwxrwxr-x  2 oracle oinstall   29 Aug 15  2009 sshsetup
drwxr-xr-x 14 oracle oinstall 4096 Aug 15  2009 stage
-rw-r--r--  1 oracle oinstall 5402 Aug 17  2009 welcome.html
[oracle@ip-172-31-45-235 database]$ █

```

```

drwxrwxr-x  2 oracle oinstall  29 Aug 15  2009 sshsetup
drwxr-xr-x 14 oracle oinstall 4096 Aug 15  2009 stage
-rw-r--r--  1 oracle oinstall 5402 Aug 17  2009 welcome.html
[oracle@ip-172-31-45-235 database]$ ./runInstaller &
[1] 17888
[oracle@ip-172-31-45-235 database]$ Starting Oracle Universal Installer...
Checking Temp space: must be greater than 120 MB.  Actual 94944 MB      Passed
Checking swap space: 0 MB available, 150 MB required.  Failed <<<
Checking monitor: must be configured to display at least 256 colors.  Actual 16777216      Passed
Some requirement checks failed. You must fulfill these requirements before
continuing with the installation,
Continue? (y/n) [n]

```

Step12:We need to add swap space at least 4G

```

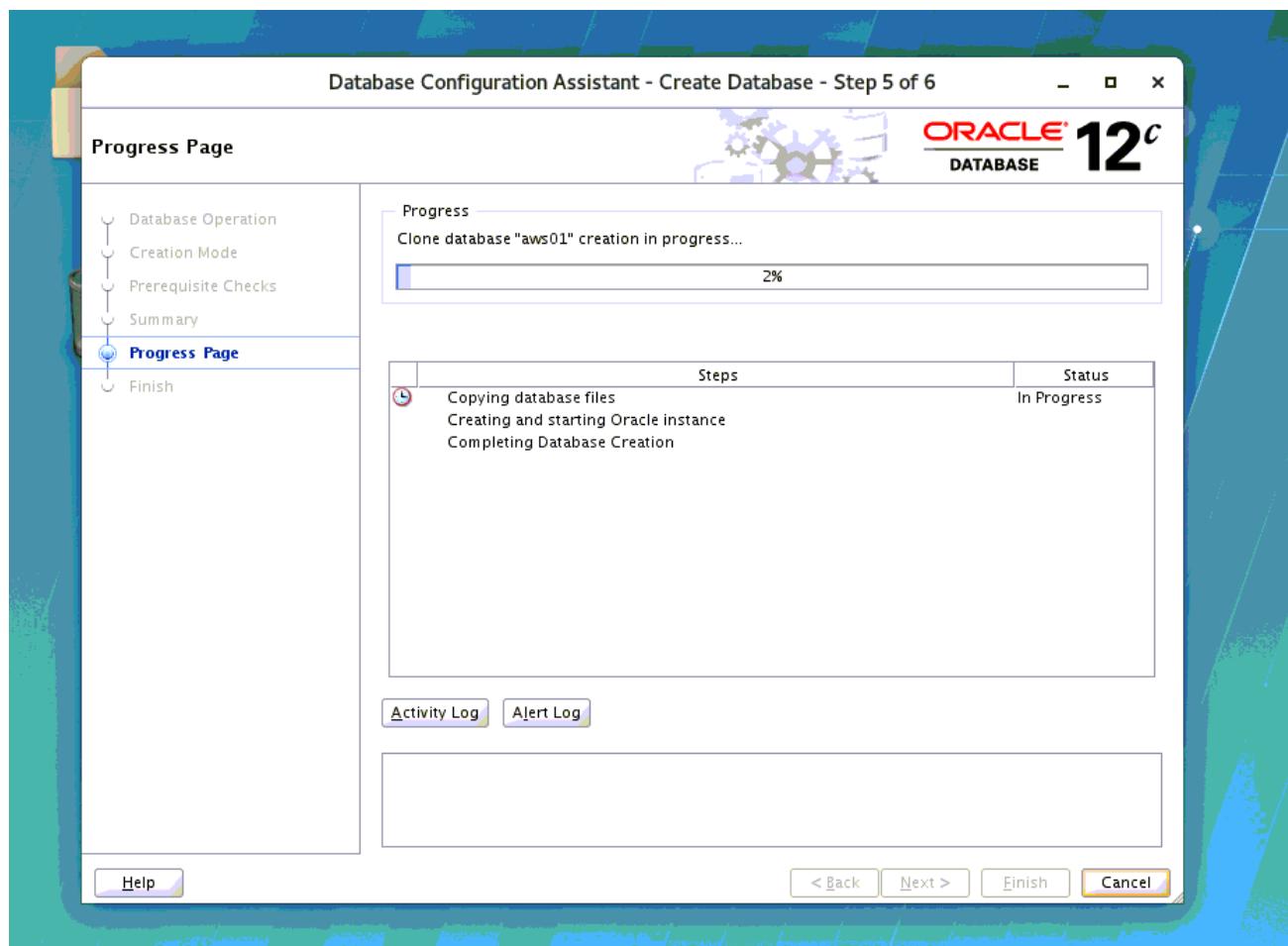
[root@ip-172-31-45-235 ~]#
[root@ip-172-31-45-235 ~]# dd if=/dev/zero of=/swapfile bs=1G count=8
3+0 records in
3+0 records out
3589934592 bytes (8.6 GB) copied, 123.824 s, 69.4 MB/s
[root@ip-172-31-45-235 ~]# chmod 600 /swapfile
[root@ip-172-31-45-235 ~]# mkswap /swapfile
Setting up swapspace version 1, size = 8388604 KiB
no label, UUID=f801b12c-3c52-4370-9e16-0d17bb24c753
[root@ip-172-31-45-235 ~]# swapon /swapfile
[root@ip-172-31-45-235 ~]# swapon -s
Filename              Type      Size    Used    Priority
/swapfile            file     8388604  0       -2
[root@ip-172-31-45-235 ~]# vi /etc/fstab
[root@ip-172-31-45-235 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Wed Oct 17 12:18:06 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=88fd4d41-c180-4721-80c6-535249a4a2bb /           xfs        defaults        0 0
/swapfile swap swap defaults 0 0
[root@ip-172-31-45-235 ~]#

```

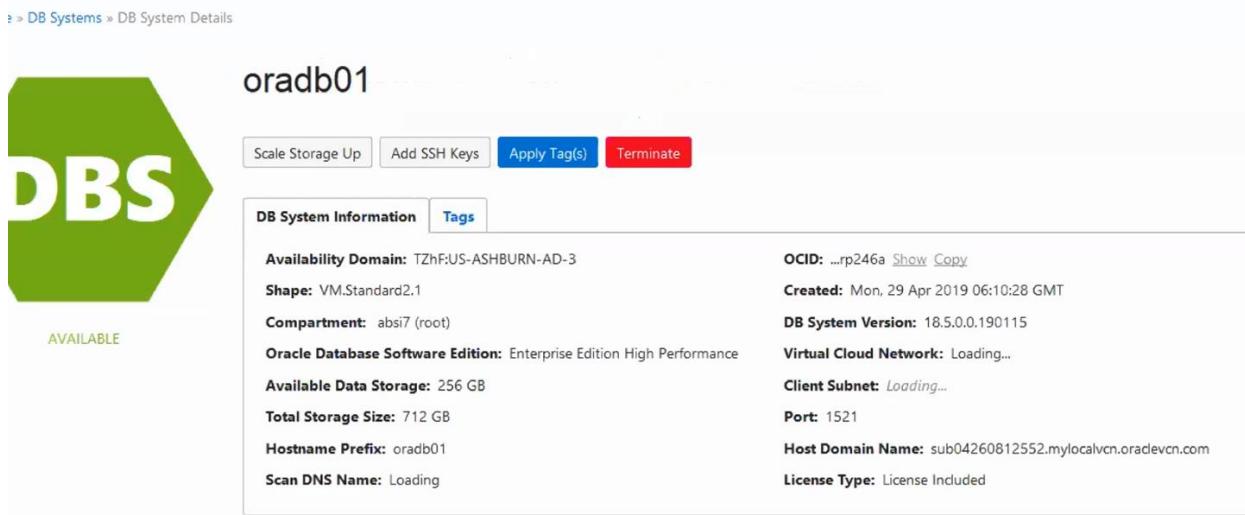
Step13: Successfully oracle DB is uploaded inside the ec2 linux server. Further database creation will be uploaded in the successive time lines.

Facing the challenge to change from Oracle 11G to 12c since we couldn't find the compatibility with the prerequisites done so we changed from 11G to 12c.



Process 2:

Initially, oracle virtual machine database instance was created as shown in fig.1 but since it stopped its connectivity with its VM we were in last minute change from VM database to oracle database service as in Fig.2.



The screenshot shows the Oracle Database Service (DBS) details for a database named 'oradb01'. The interface includes a large green arrow icon with 'DBS' on it, indicating the service type. The database status is listed as 'AVAILABLE'. On the right, there are several tabs: 'DB System Information' (selected), 'Tags', 'Logs', 'Metrics', 'Logs', 'Logs', and 'Logs'. Below the tabs, detailed information is provided:

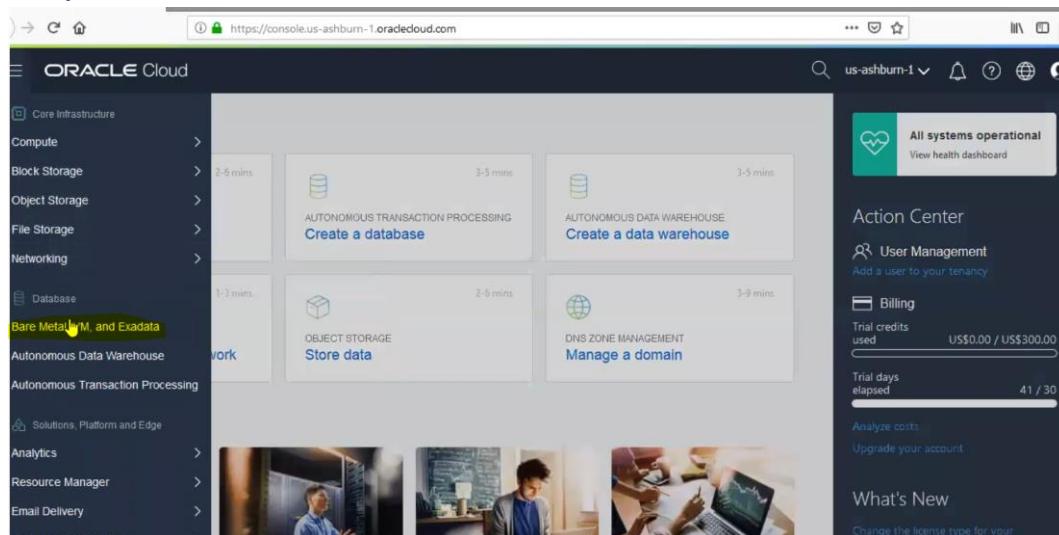
DB System Information	Tags
Availability Domain: TZhF:US-ASHBURN-AD-3	OCID: ...rp246a Show Copy
Shape: VM.Standard2.1	Created: Mon, 29 Apr 2019 06:10:28 GMT
Compartment: absi7 (root)	DB System Version: 18.5.0.0.190115
Oracle Database Software Edition: Enterprise Edition High Performance	Virtual Cloud Network: Loading...
Available Data Storage: 256 GB	Client Subnet: Loading...
Total Storage Size: 712 GB	Port: 1521
Hostname Prefix: oradb01	Host Domain Name: sub04260812552.mylocalvcn.oraclevcn.com
Scan DNS Name: Loading	License Type: License Included

At the top of the page, there are buttons for 'Scale Storage Up', 'Add SSH Keys', 'Apply Tag(s)', and 'Terminate'.

Fig1

Creating oracle Database service:

Step1:



The screenshot shows the Oracle Cloud Core Infrastructure dashboard. The left sidebar lists services under 'Core Infrastructure': Compute, Block Storage, Object Storage, File Storage, Networking, Database (selected), Bare Metal VM, and Exadata, Autonomous Data Warehouse, Autonomous Transaction Processing, Solutions, Platform and Edge, Analytics, Resource Manager, Email Delivery, and Application Integration. The main area displays four service cards: 'AUTONOMOUS TRANSACTION PROCESSING' (Create a database, 2-5 mins), 'AUTONOMOUS DATA WAREHOUSE' (Create a data warehouse, 3-5 mins), 'OBJECT STORAGE' (Store data, 1-3 mins), and 'DNS ZONE MANAGEMENT' (Manage a domain, 3-9 mins). To the right, there's an 'Action Center' with sections for User Management, Billing (trial credits used: US\$0.00 / US\$300.00, trial days elapsed: 41 / 30), and What's New.

Fig2

Step2: Giving the credentials required so as to create the database systems.

The screenshot shows the Oracle Cloud interface for creating a new DB System. On the left, a sidebar lists 'Bare Metal, VM, and Exadata' options, with 'DB Systems' selected. The main panel is titled 'Create DB System' and shows configuration steps. Step 1: 'SELECT A DB SYSTEM' has 'VM' selected. Step 2: 'SELECT A DB SYSTEM' shows 'VM.Standard2.1' selected. Step 3: 'SELECT ORACLE DATABASE SOFTWARE EDITION' shows 'Enterprise Edition High Performance' selected. Step 4: 'SELECT A STORAGE SIZE' is partially visible. Step 5: 'LICENSE TYPE' shows 'LICENSE INCLUDED' selected. Step 6: 'CREATE' button at the bottom. To the right, a preview window shows the DB system details: 'us-ashburn-1', 'Launched: Mon, 29 Apr 2019 06:10:28 GMT', and 'Displaying 1 DB Systems'.

This screenshot continues the DB System creation process. Step 7: 'SELECT DATABASE WORKLOAD' shows 'ON-LINE TRANSACTION PROCESSING (OLTP)' selected. Step 8: 'TAGS' section is present. At the bottom, a large green box highlights the 'Launch DB System' button. To the right, the preview window remains the same, showing the launched DB system.

Step3: The database provisioning time after creating the database system in oracle takes at least 3 hours and then the database system is created as shown in Fig5.

are Metal, VM, and xadata

DB Systems

- Standalone Backups

Scope

COMPARTMENT

absi7 (root) ▼

Don't see what you're looking for? ?

DB Systems in absi7 (root) Compartment

Displaying 2 DB Systems

Launch DB System				
 oradb01	Availability Domain: TZhF:US-ASHBURN-AD-3 OCID: ...u3d7ia Show Copy	Oracle Database Software Edition: Enterprise Edition Extreme Performance Shape: VM.Standard2.1	Virtual Cloud Network: my-local-vcn Client Subnet: Public Subnet TZhF:US-ASHBURN-AD-3 Private IP: Loading... Public IP: Loading... Available Data Storage: 256 GB Total Storage Size: 712 GB	Launched: Mon, 29 Apr 2019 16:13:59 GMT
 oradb01	Availability Domain: TZhF:US-ASHBURN-AD-3 OCID: ...rp246a Show Copy	DB System Version: 18.5.0.0.190115 Oracle Database Software Edition: Enterprise Edition High	Virtual Cloud Network: my-local-vcn Client Subnet: Public Subnet TZhF:US-ASHBURN-AD-3	Launched: Mon, 29 Apr 2019 06:10:28 GMT

Fig4

Bare Metal, VM, and Exadata

DB Systems

- Standalone Backups

Scope

COMPARTMENT

absi7 (root) ▼

Don't see what you're looking for? ?

DB Systems in absi7 (root) Compartment

Displaying 1 DB Systems

Launch DB System			
 oradb01	DB System Version: 18.5.0.0.190115 Oracle Database Software Edition: Enterprise Edition Extreme Performance Shape: VM.Standard2.1	Virtual Cloud Network: Loading... Client Subnet: Public Subnet TZhF:US-ASHBURN-AD-3 Private IP: 10.0.2.6 Public IP: 132.145.216.239 Available Data Storage: 256 GB Total Storage Size: 712 GB	Launched: Mon, 29 Apr 2019 16:13:59 GMT

Fig5

Step4: After creating the Database inside the ORACLE cloud service and AWS EC2 we need to mount the database into the Oracle database of 7.3M records .

```
^Cselect count(*) from EMP1
      *
ERROR at line 1:
ORA-01013: user requested cancel of current operation

SQL> drop table EMP1;
Table dropped.

SQL> drop table emp;
Table dropped.

SQL>
SQL> insert into test (select * from test);
917504 rows created.

SQL> insert into test (select * from test);
1835008 rows created.

SQL> insert into test (select * from test);
3670016 rows created.

SQL> insert into test (select * from test);
7340032 rows created.

SQL> ■
```



```
SQL> insert into test (select * from test);
57344 rows created.

SQL> insert into test (select * from test);
114688 rows created.

SQL> insert into test (select * from test);
229376 rows created.

SQL> insert into test (select * from test);
458752 rows created.

SQL> insert into test (select * from test);
917504 rows created.

SQL> insert into test (select * from test);
1835008 rows created.

SQL> insert into test (select * from test);
3670016 rows created.

SQL> insert into test (select * from test);
7340032 rows created.

SQL>
```

Step6: The database insertion with write , read ,commit and transaction queries should be run infinite times later.

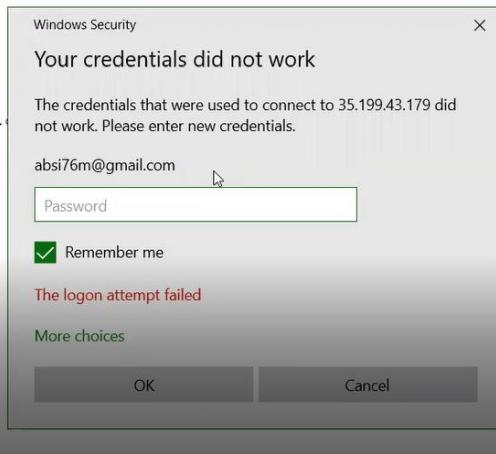
Process 3:

Creating an account in GCP and opening the windows remote desktop.

This is done because to download solar winds inside the windows server and connect the databases to check its performance.

First troubleshooting issues: I cannot get the remote desktop connection in GCP through instance.

```
23 Username:gssgkb
24 Pwd:2019Password!
25
26
27 Holy77#*
28
29
30 Oracle Cloud:
31 User name: aoer5b@mail.umkc.edu
32 Cloud name: absi7
33 Password: Cloud Computing 2019
34
35
36
37 Public DNS
38 ec2-3-83-112-214.compute-1.amazonaws.
39 User name
40 Administrator
41 Password
42 S=iggCo4FE
43
44
45
46
47
48
49
50 Win Mon
51
52 user = administrator
53 pass = 6(?DQQsL^5//R58
54
55
56
57 W!QHL4_~Eq?,NYS
58
59 ^yG-o8N-n8\Bv,W
```



References using some you tubes videos on how to open remote desktop in GCP and some links as follows:

https://cloud.google.com/compute/docs/instances/connecting-to-instance

ud Why Google Solutions Products Pricing Getting started

cts

Compute Engine > Documentation

Connecting to instances

This page describes some of the most common ways to connect to your Compute Engine Windows instances.

For additional ways to connect to your instances, see the following:

- [Connecting using third-party tools](#), including PuTTY.
- [Connecting to instances that have no external IP addresses](#)

Before you connect, you must set up user access to your instance. This page also describes how to [use a Linux VM guide](#) or the [Quickstart using a Windows VM guide](#) to create your default user access.

Complete at least one of those guides before continuing.

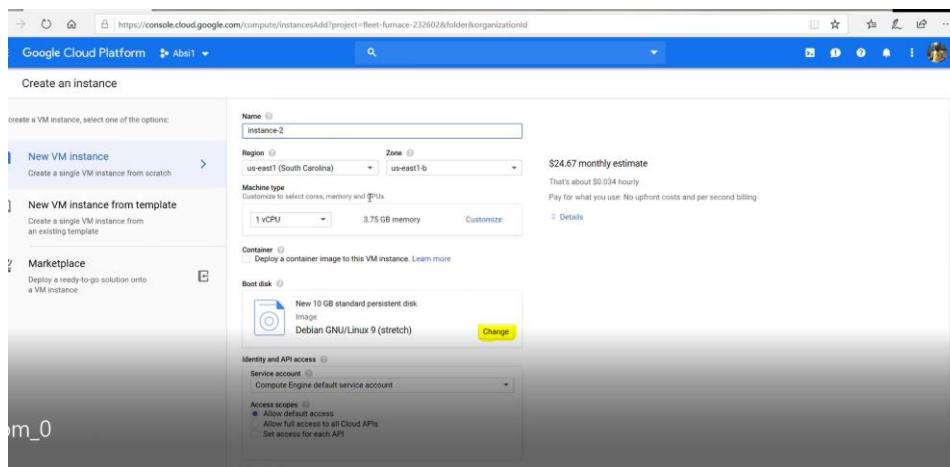
To learn more about managing user access to your instance, see [Managing Instance Access](#).

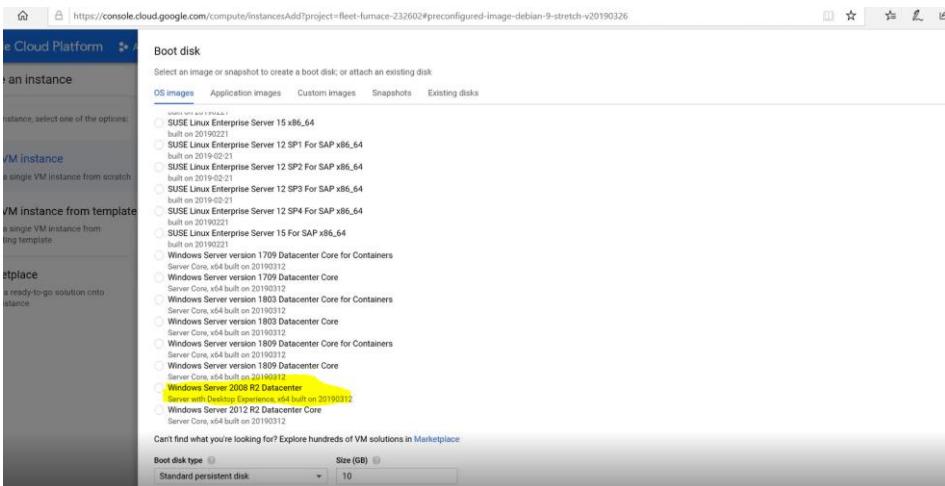
Connecting to Linux instances

To connect to Linux instances through the Google Cloud Platform Console or the command line, follow the steps in the tabs below:

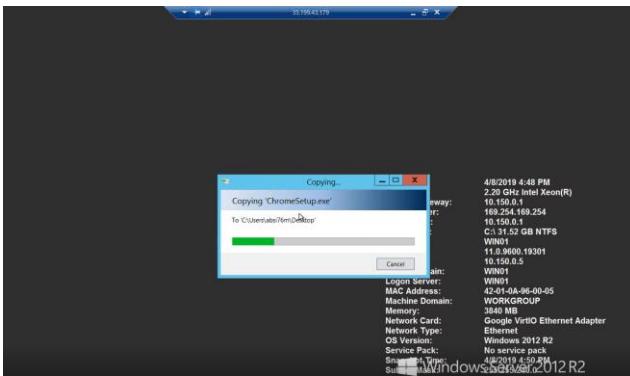
If these basic SSH options do not work for you, you might need to connect to instances using a third-party tool like PuTTY.

By changing the server from Debian to windows server with desktop remote access as follows:

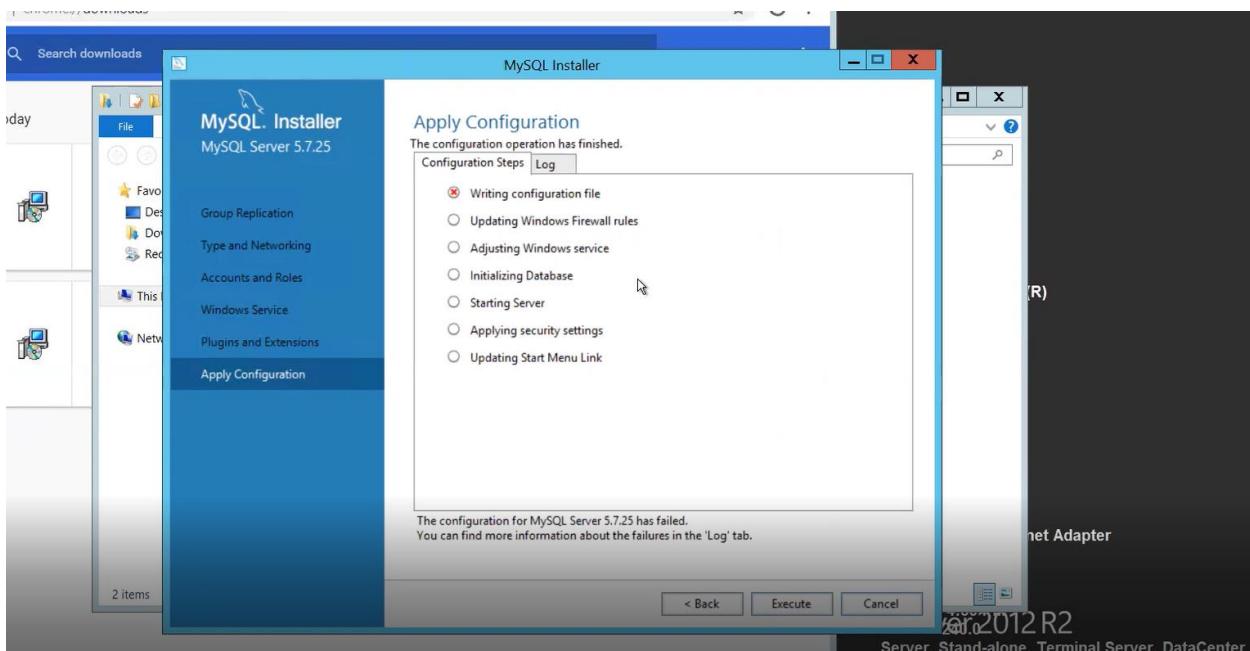




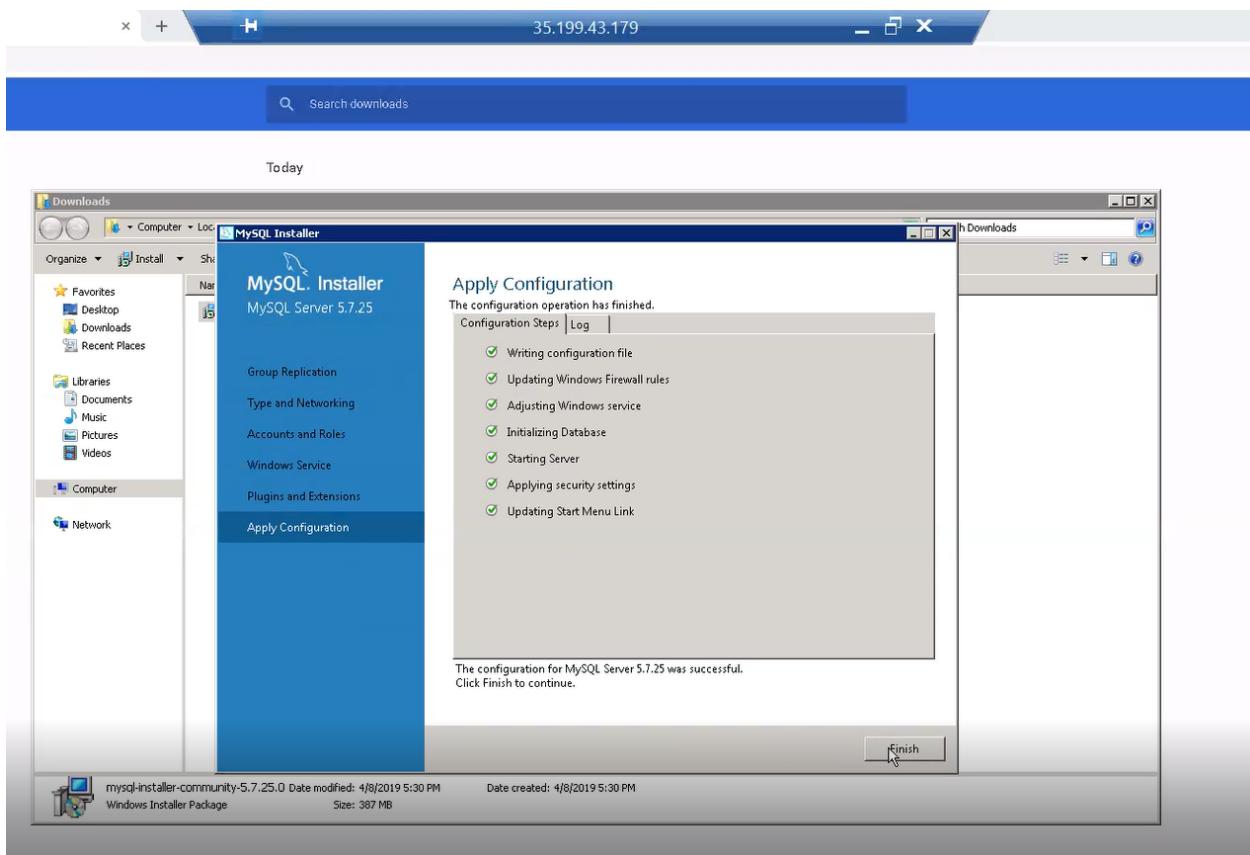
After opening the windows remote server copy the chrome setup and paste it in the cloud VM and run it since the browser in the vm does the security violation :



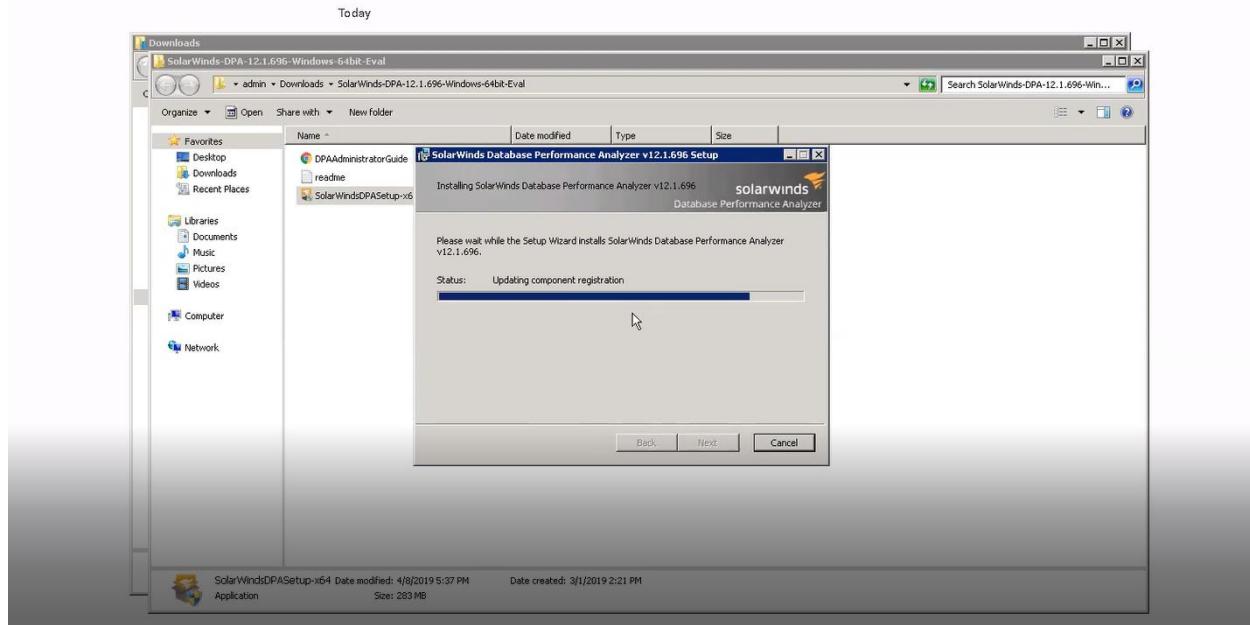
Download repository DB where the analyzed data will get stored in the solar winds repository DB i.e. MySQL repository community DB will be downloaded and while installing that in the Vm troubleshooting occurs as follows:



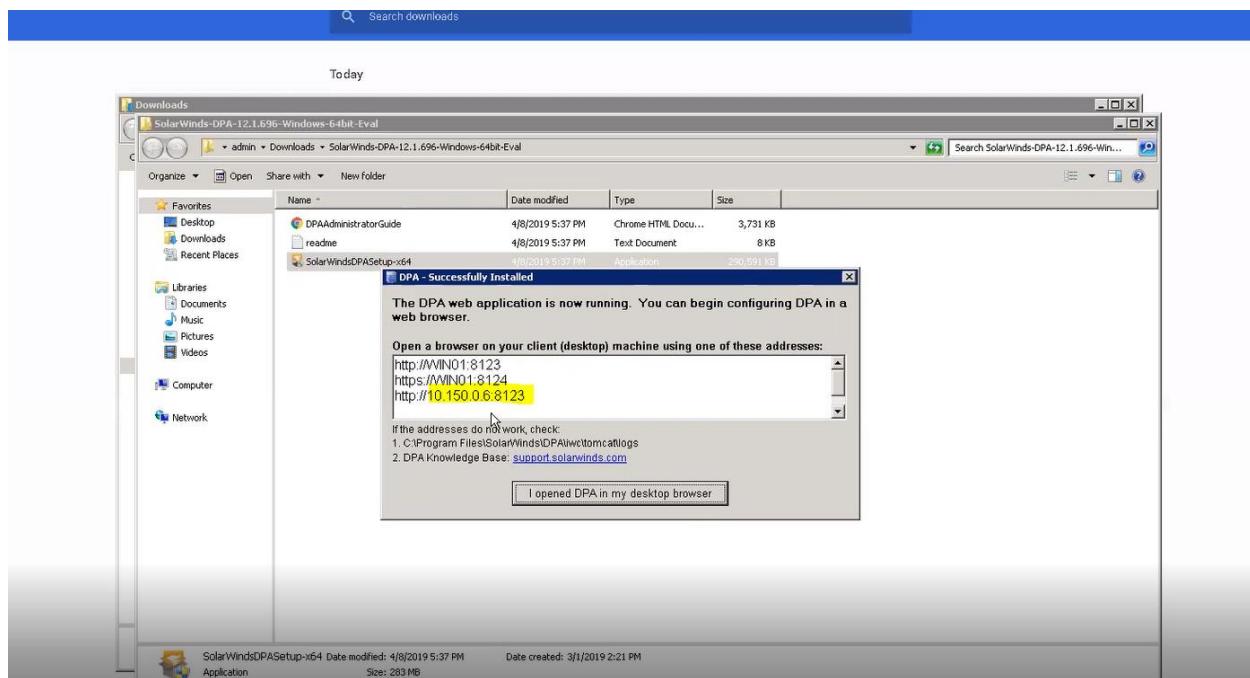
Disabling windows firewall in VM and re install the SQL repository it will be executed perfectly as follows:



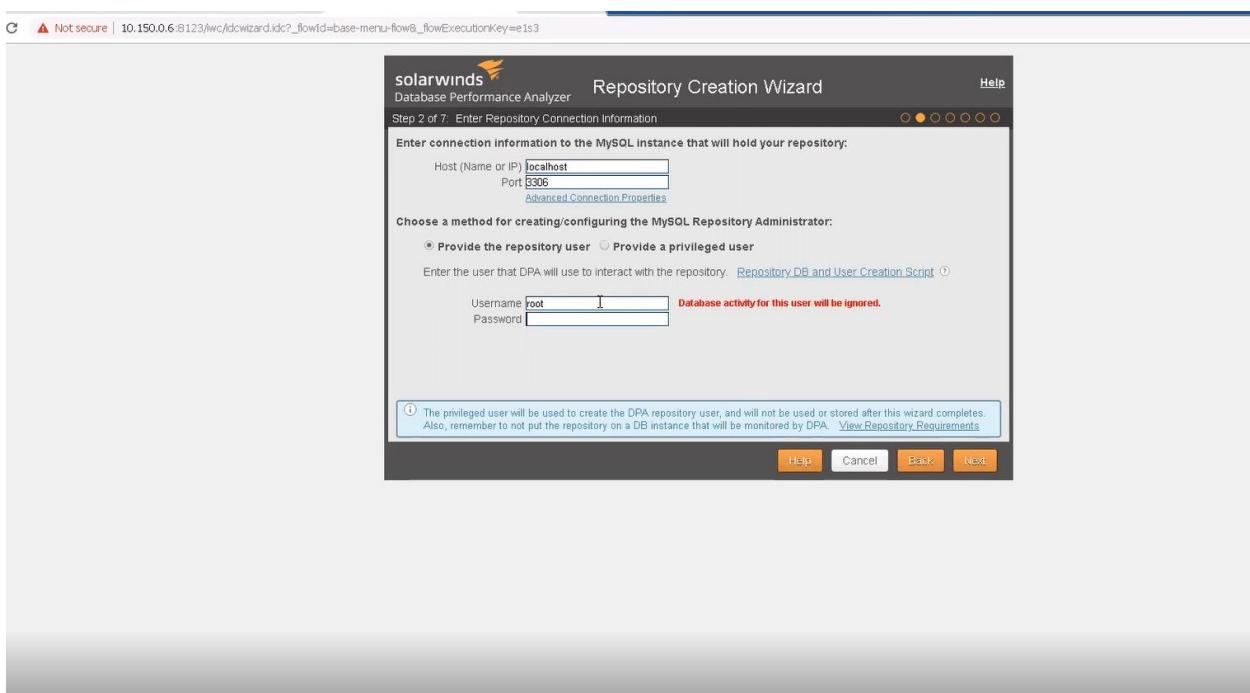
Open the google chrome and start downloading the solarwinds for oracle and start installing that in the VM.



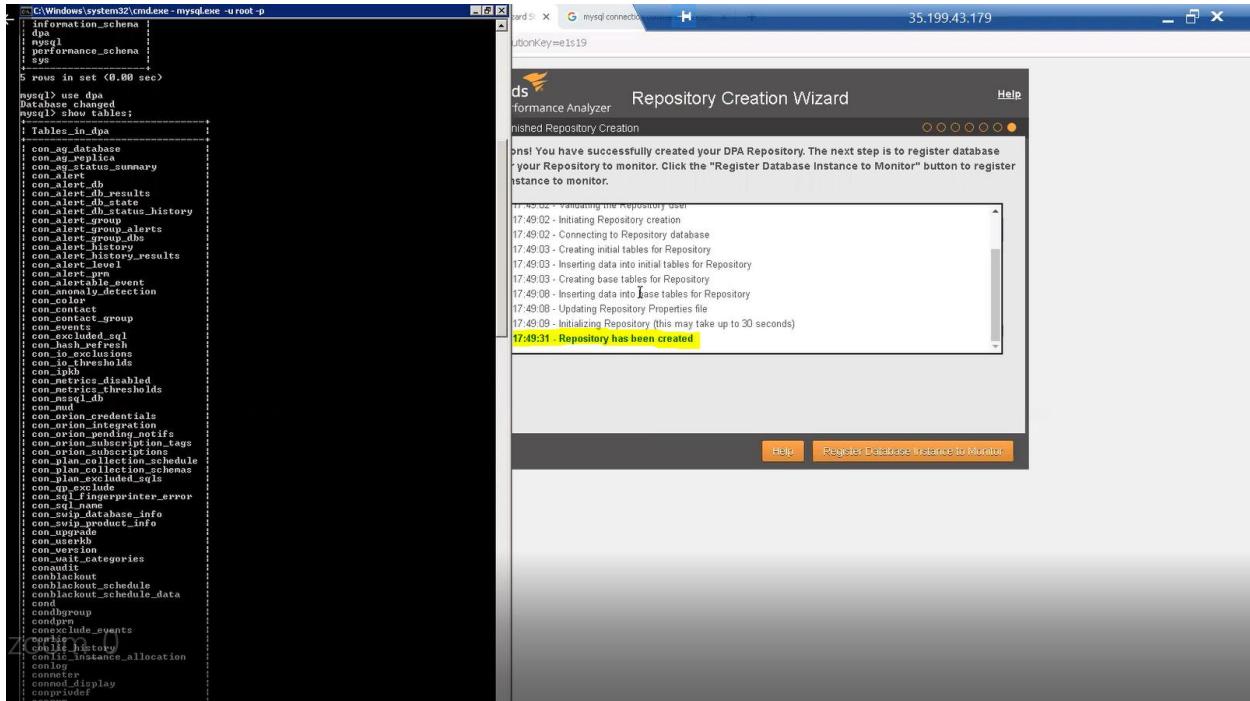
Database performance analyzer has been successfully installed in GCP VM as follows:



Make note of the port and open a browser and give the port in the browser then the solarwind will open as follows:



Using the command prompt change the directory and execute the solarwind and your Database DPA will be created inside the solarwind.



Now connect the AWS Database to GCP solarwind.

Open the AWS console and ping the GCP public IP, it will be connected as follows:

```

lsnrctl start
sqlplus as sysdba
story
p-172-31-44-178 bin]$ ./sqlplus sys as sysdba
   Release 12.1.0.2.0 Production on Mon Apr 8 17:54:01 2019
   (c) 1982, 2014, Oracle. All rights reserved.

sword:
to an idle instance.

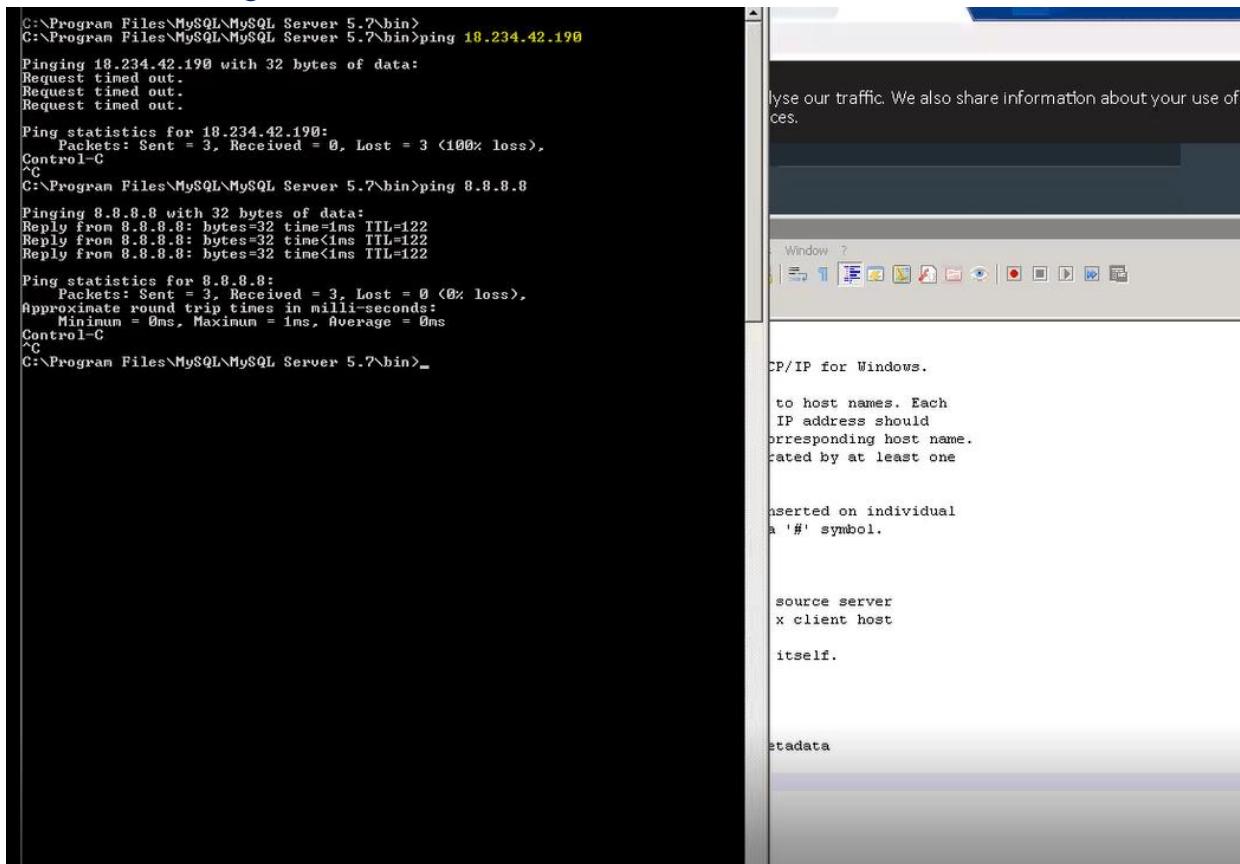
tup
stance started.

Item Global Area 2466250752 bytes
          2927384 bytes
Size          67108996 bytes
Buffers      1778384896 bytes
ers          13848576 bytes
mounted.

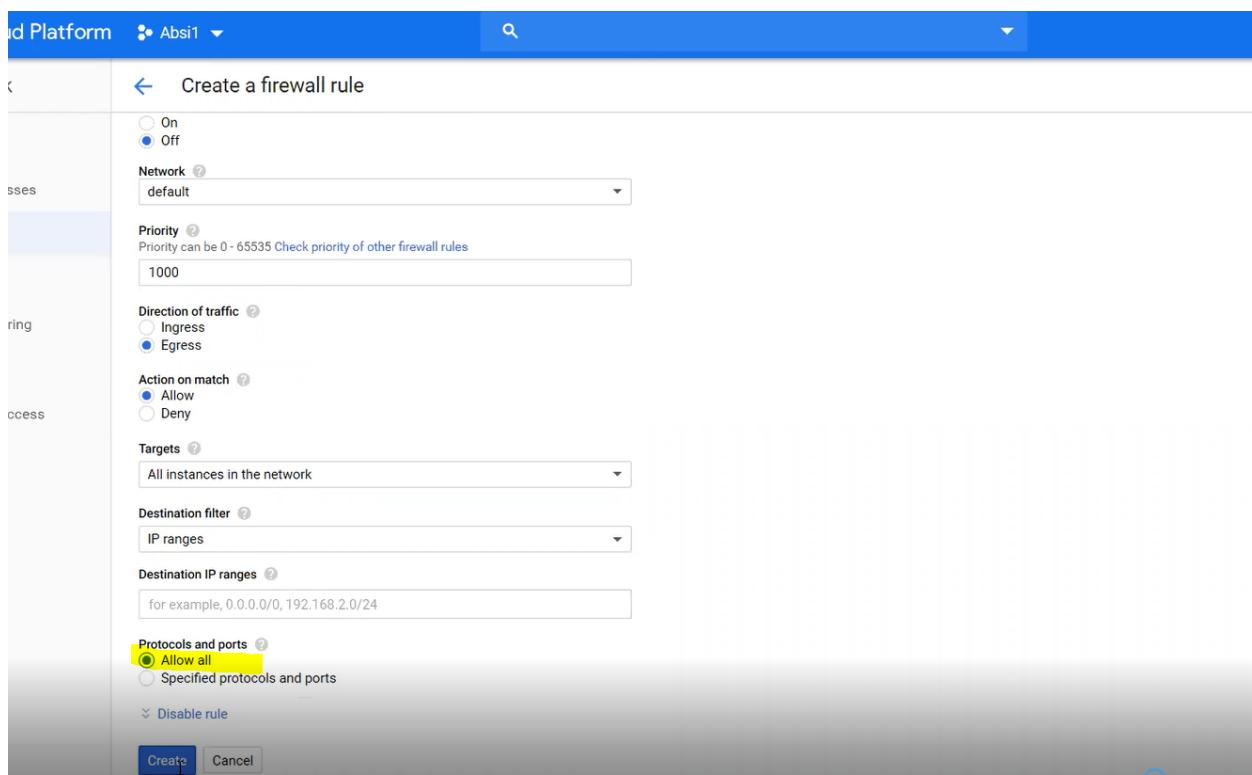
ted from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit P
Partitioning, OLAP, Advanced Analytics and Real Application Testing options
p-172-31-44-178 bin]$ 
p-172-31-44-178 bin]$ 
p-172-31-44-178 bin]$ ping 35.199.43.179
99.43.179 (35.199.43.179) 56(84) bytes of data.
from 35.199.43.179: icmp_seq=1 ttl=120 time=2.37 ms
from 35.199.43.179: icmp_seq=2 ttl=120 time=1.76 ms
from 35.199.43.179: icmp_seq=3 ttl=120 time=1.80 ms
from 35.199.43.179: icmp_seq=4 ttl=120 time=1.71 ms
9.43.179 ping statistics ---
transmitted, 4 received, 0% packet loss, time 3005ms
vg/max/mdev = 1.719/1.914/2.374/0.268 ms
p-172-31-44-178 bin]$

```

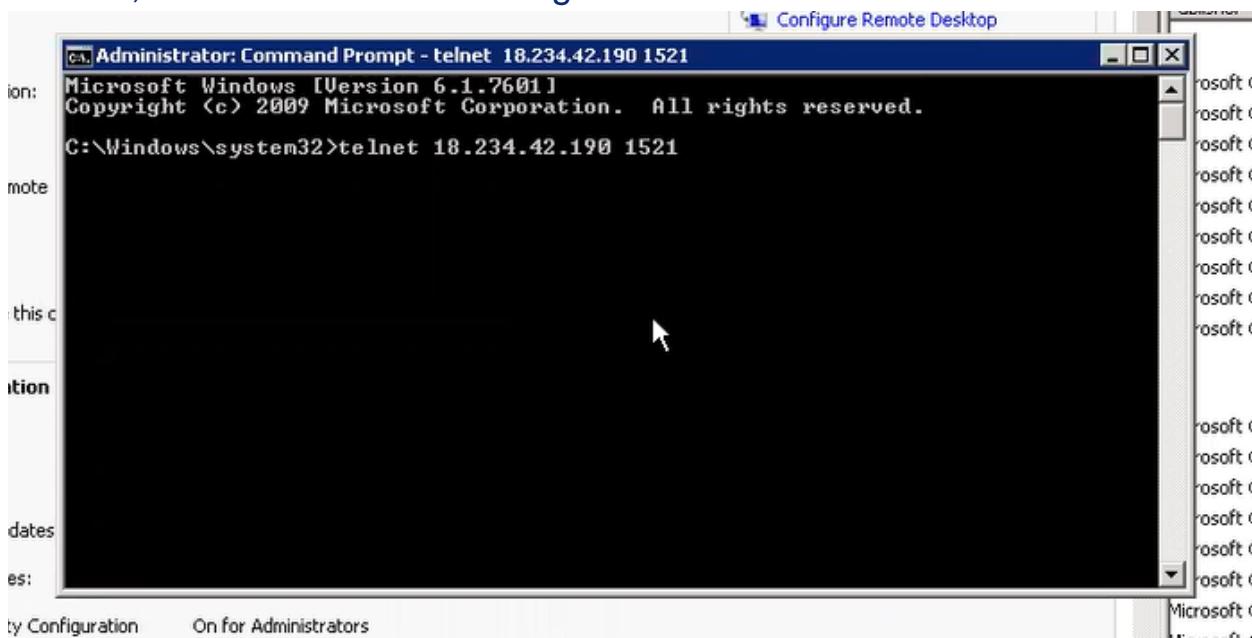
But the GCP does not ping because of firewall rules and we need to add a new rule and then configure telnet and establish the connection.



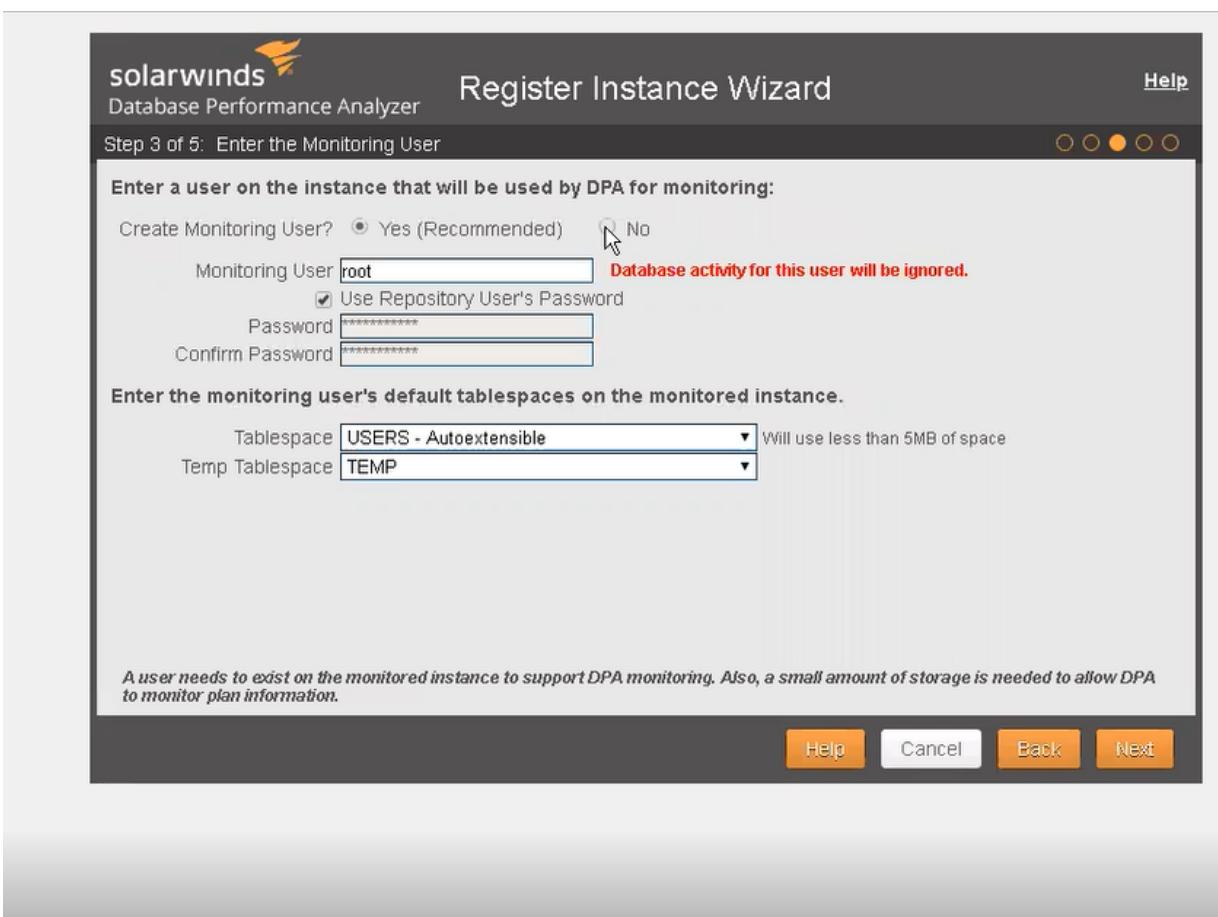
Enabling firewall rule in GCP:



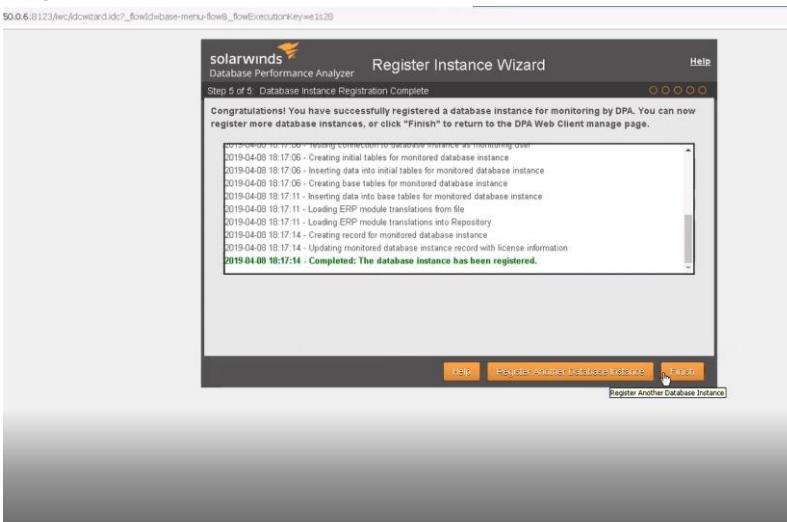
In GCP, AWS is connected through telnet as follows:



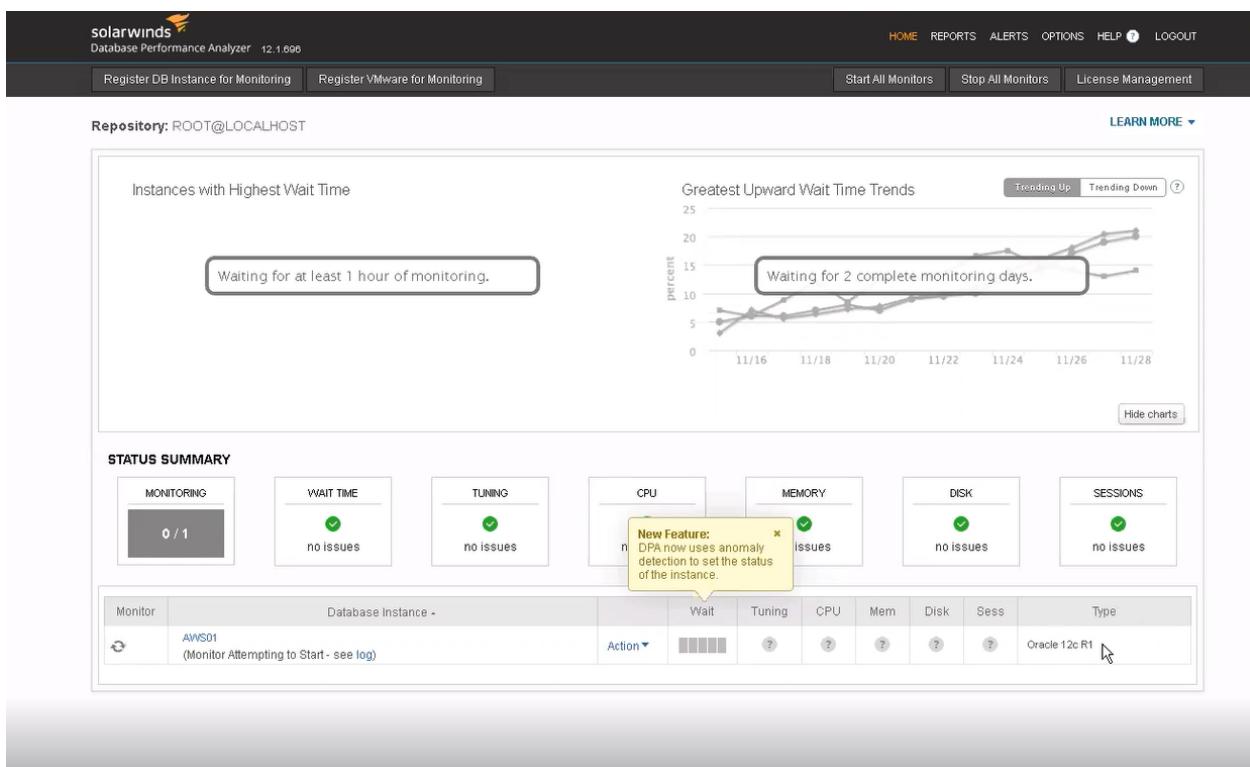
Connect the database in solar winds there comes another issues:



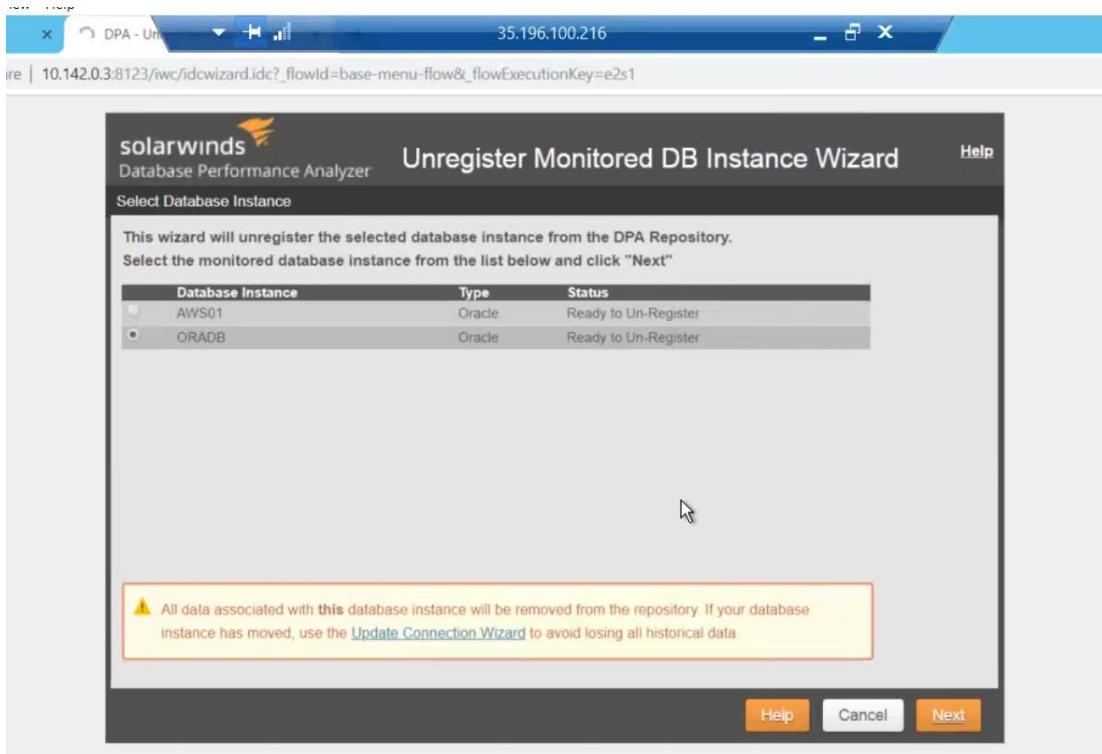
By creating the monitoring user and password AWS database has been successfully registered.



Thus solarwind is created in GCP and AWS is registered as follows:



Likewise connect the oracle database service with the solar winds as follows:



The screenshot shows a web-based database monitoring application. At the top left is a search bar with the placeholder "Type DB instance name" and a "Search" button. To the right are icons for "Groups" and "Settings". The main area is a table with the following data:

Monitor	Database Instance		Wait	Tuning	CPU	Mem	Disk	Sess	Type
↑ ON	AWS01 (Running)	Action ▾	<div style="width: 100%;"> </div>	!	✓	✓	✓	✓	Oracle 12c R1
↑ ON	ORACLE01 (Running)	Action ▾	<div style="width: 100%;"> </div>	!	✓	✓	✓	✓	Oracle 11g R2

Process4: Iteration:

After successfully created the Databases and connecting them with the GCP desktop we need to start the iterations to observe the performance evaluation by the solarwinds evaluation tool.

The screenshot shows two side-by-side Oracle SQL command windows. The left window displays the creation and dropping of tables (EMP1 and emp) and the execution of several INSERT INTO TEST statements, each selecting all columns from the TEST table. The right window shows a similar sequence of operations, with the first few lines indicating an error (ORA-01013: user requested cancel of current operation) and subsequent lines showing successful insertions of 57344, 114688, 229376, 458752, 917504, 1835008, 3670016, 7340032 rows respectively. Both windows end with a SQL> prompt.

```
^Cselect count(*) from EMP1
      *
ERROR at line 1:
ORA-01013: user requested cancel of current operation

SQL> drop table EMP1;
Table dropped.

SQL> drop table emp;
Table dropped.

SQL>
SQL> insert into test (select * from test);
917504 rows created.

SQL> insert into test (select * from test);
1835008 rows created.

SQL> insert into test (select * from test);
3670016 rows created.

SQL> insert into test (select * from test);
7340032 rows created.

SQL>
```

```
SQL> insert into test (select * from test);
57344 rows created.

SQL> insert into test (select * from test);
114688 rows created.

SQL> insert into test (select * from test);
229376 rows created.

SQL> insert into test (select * from test);
458752 rows created.

SQL> insert into test (select * from test);
917504 rows created.

SQL> insert into test (select * from test);
1835008 rows created.

SQL> insert into test (select * from test);
3670016 rows created.

SQL> insert into test (select * from test);
7340032 rows created.

SQL>
```

At least 1580 times the iterations are run by us for the past weeks so that we can Evaluate their performance based on their parameters which impact such as write, read, commit and transactions.

Step2: At first the analysis were favoring the oracle database than AWS for 2 days then AWS was standard enough to beat the performance of oracle cloud DB and was taking very low time for all the iterations.

Analysis of Read, write graphs of both the databases:



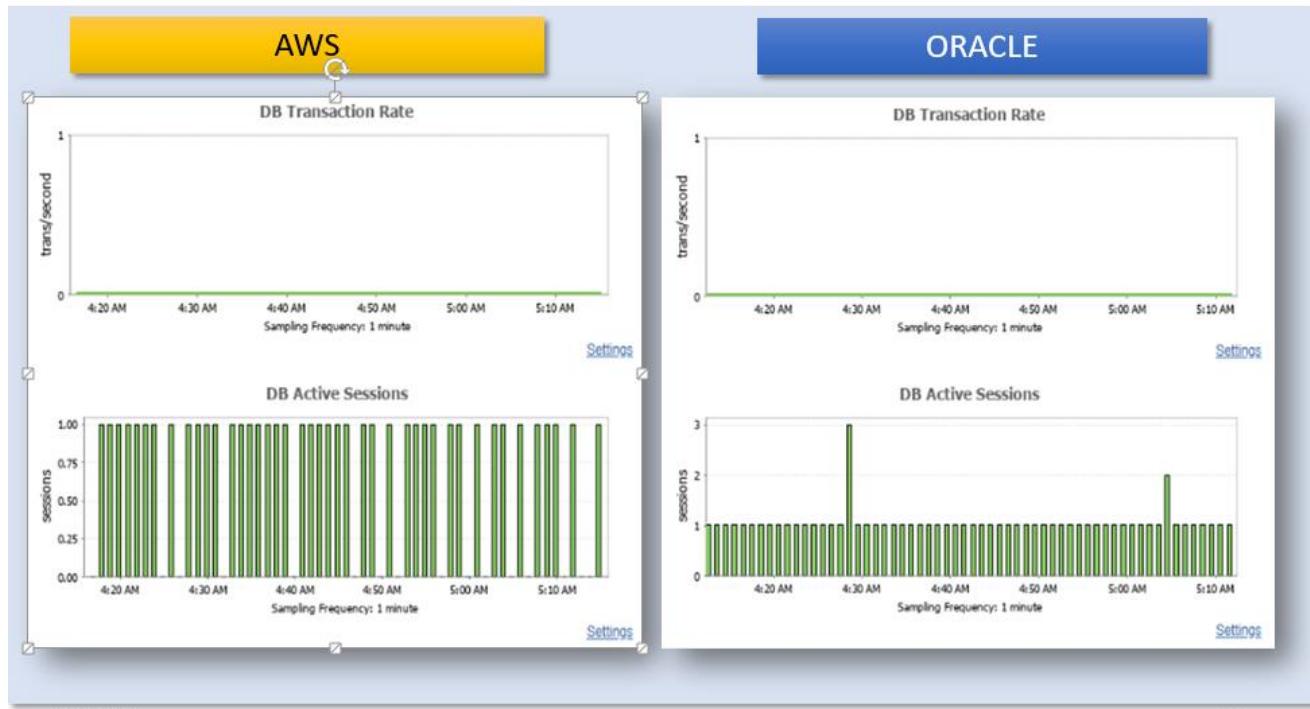
Analysis of commit time:



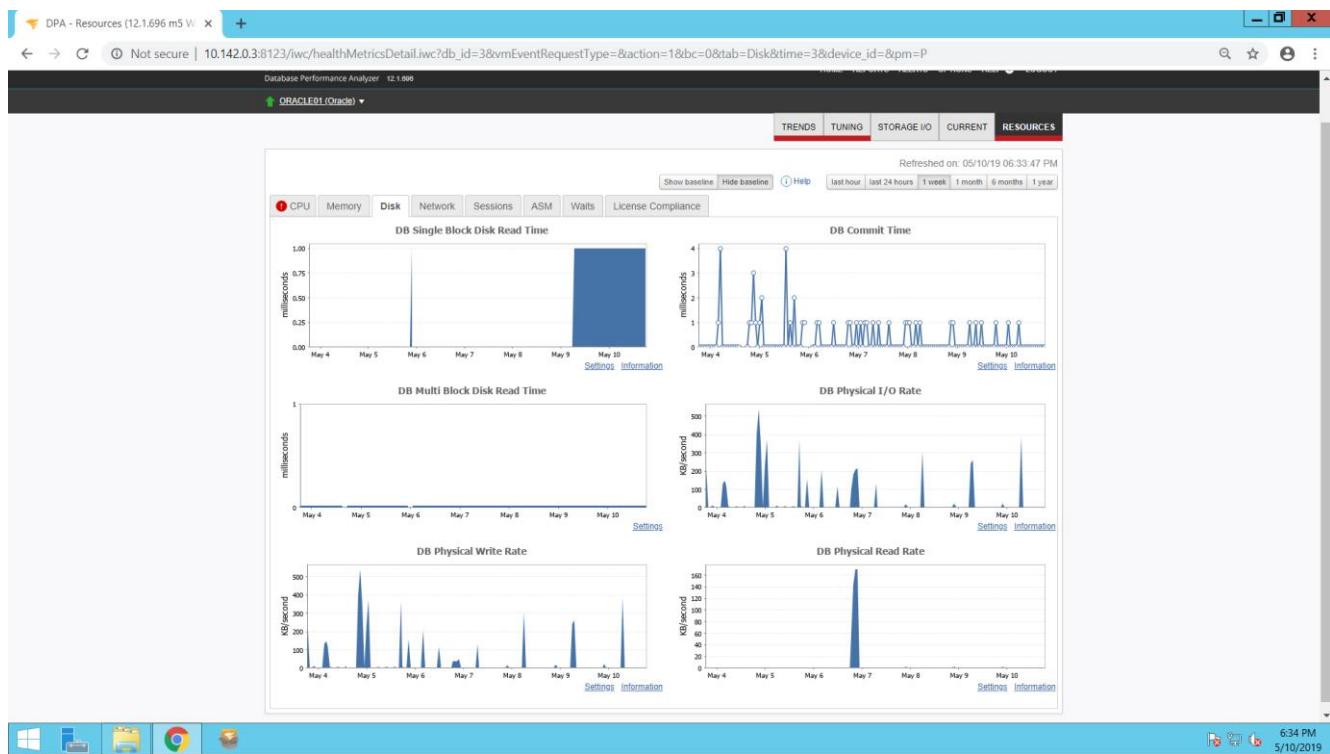
In the above figure we can clearly see how the performance rate of AWS is much better with less time than oracle cloud database.

Analysis of transactions of both the databases:

TRANSACTION

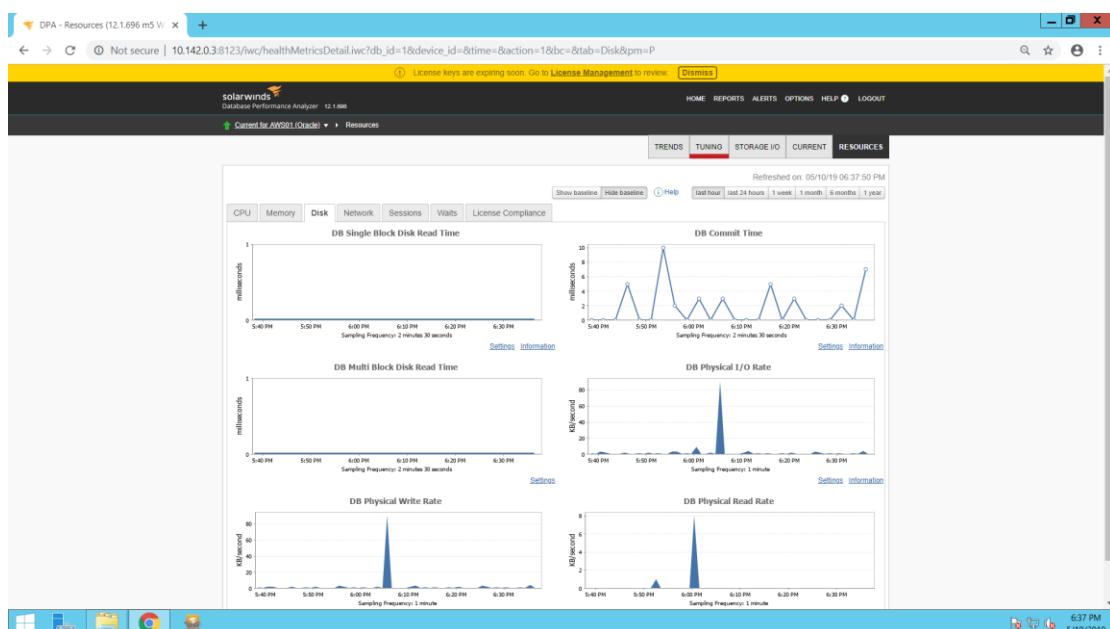


Final analysis as on 5th march 2019: Oracle DB in Solar winds:

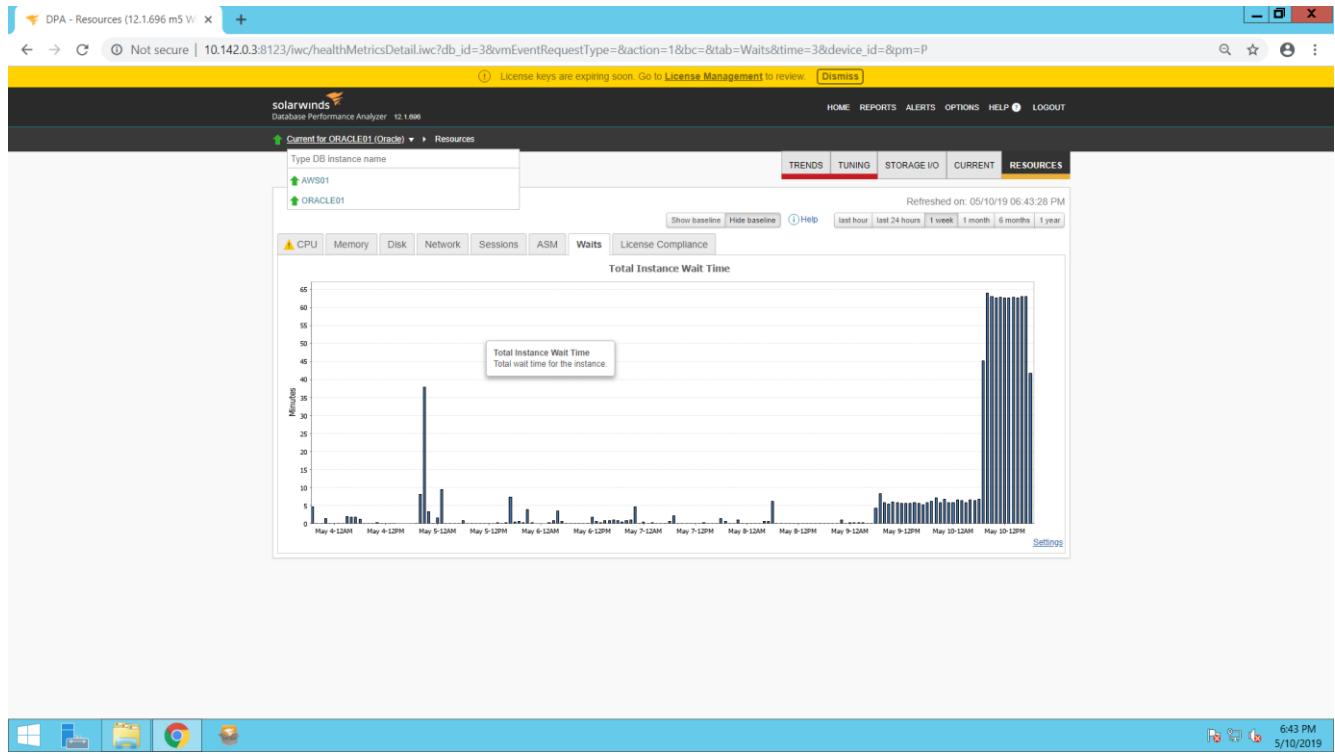


Here you can see how the read and write of oracle are highly increased.

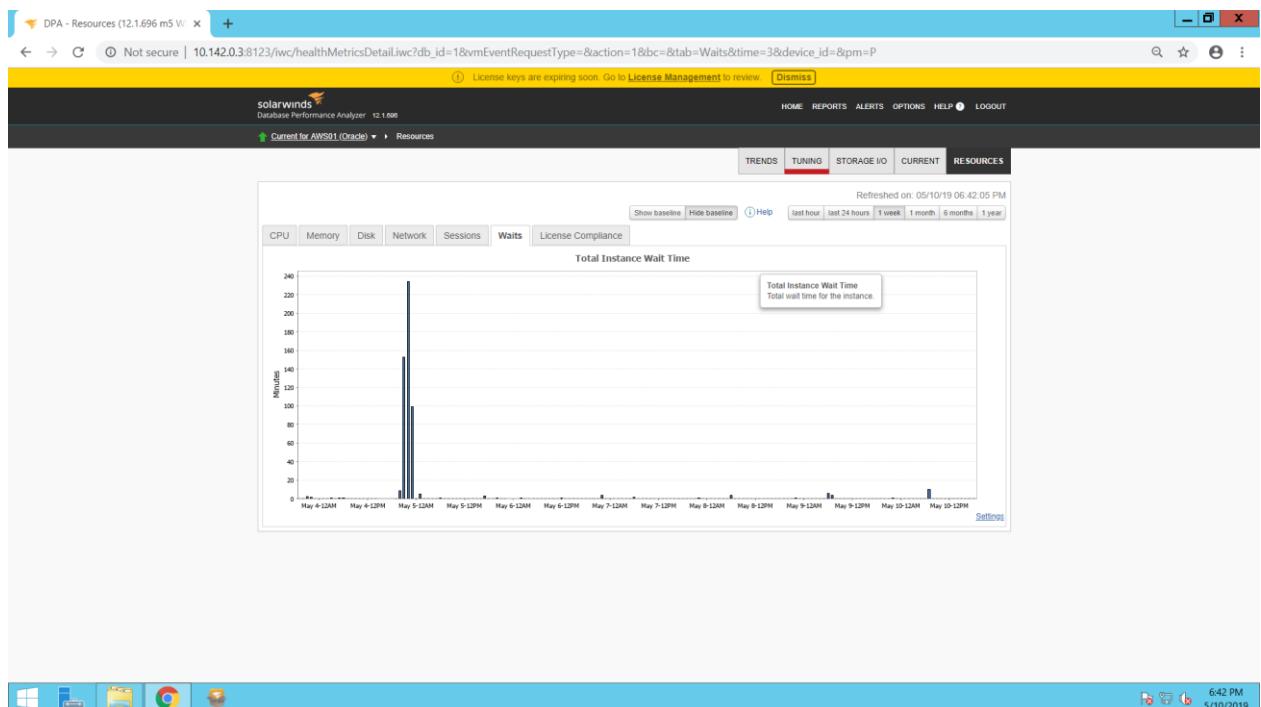
AWS was much lower than oracle:



Total waits of Oracle:



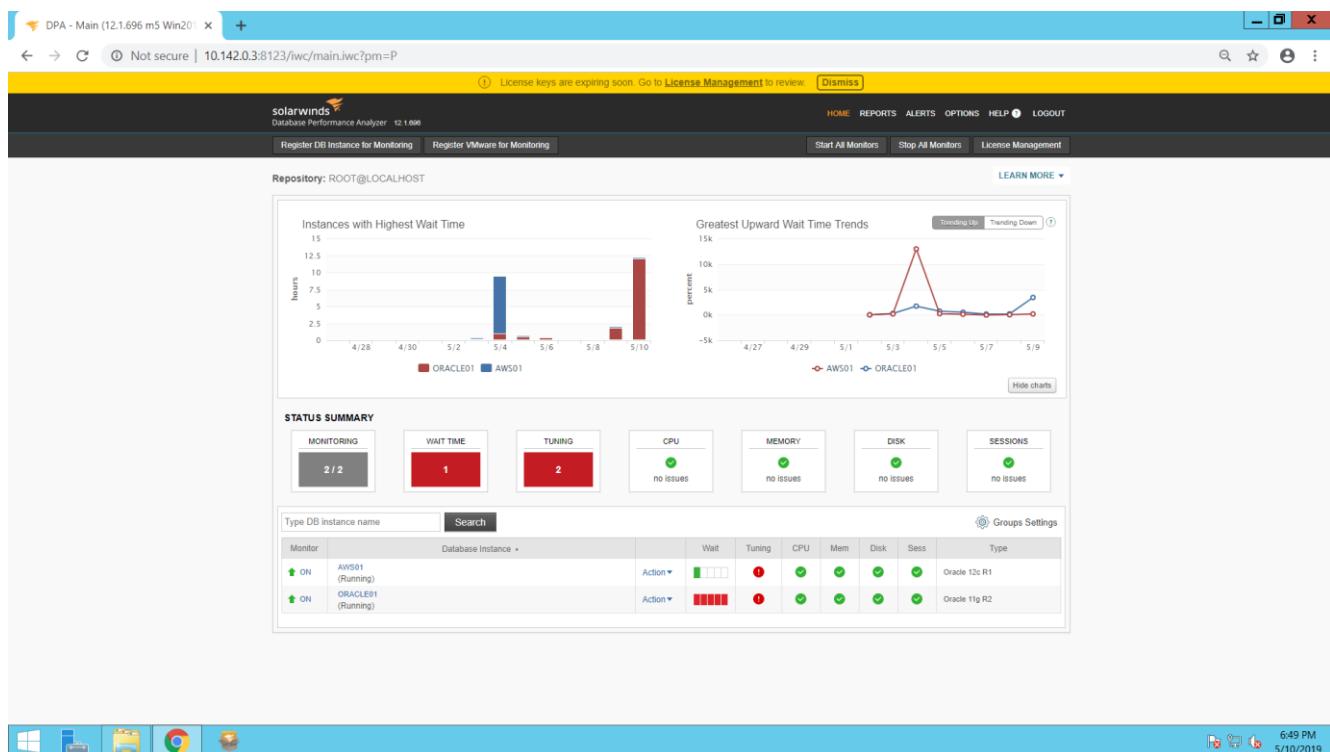
Total waits of AWS:



If you compare the above graph that ORACLE was having long waits than AWS.

Conclusion:

The Virtual machine of AWS EC2 database was much better in long run than the ORACLE database service in ways of CPU utilization, commit time, transactions and more wait time overall performance was favoring initially to ORACLE cloud but with long run AWS was much better than oracle.



The above overall performance graph of AWS and ORACLE clearly the performance difference between them.