<CUSTOMER> | EXACS Oracle Dynamic Scaling Engine

TOPICS

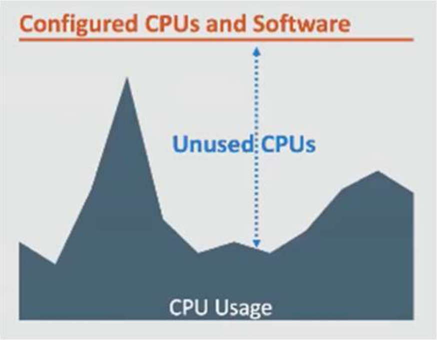
1. CPU SCALING OVERVIEW
2. MANUAL SCALE UP/DOWN OPERATIONS
3. DYNAMIC SCALE UP/DOWN OPERATIONS

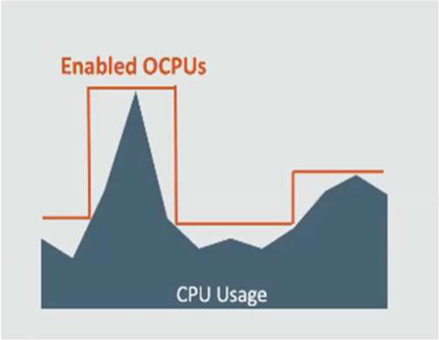
1. CPU SCALING OVERVIEW

SCALING VS COST

CPU configuration:

- On-prem (Static) . Purchase CPUs and software licenses for highest project peak load.
- Cloud (Elastic). Adjust Enabled OCPUs to match actual load CPU.





One of the greatest benefits of cloud computing is the ability to pay for only what you use. With **Oracle Autonomous Database**, the system automatically scales up and down the resources required (auto scale feature) to meet the requirements of the workload and you truly only pay for what is you use.

Many users of **Exadata Cloud Service** have asked how they can also automatically scale their system in a similar manner, sizing it continuously to meet demand.

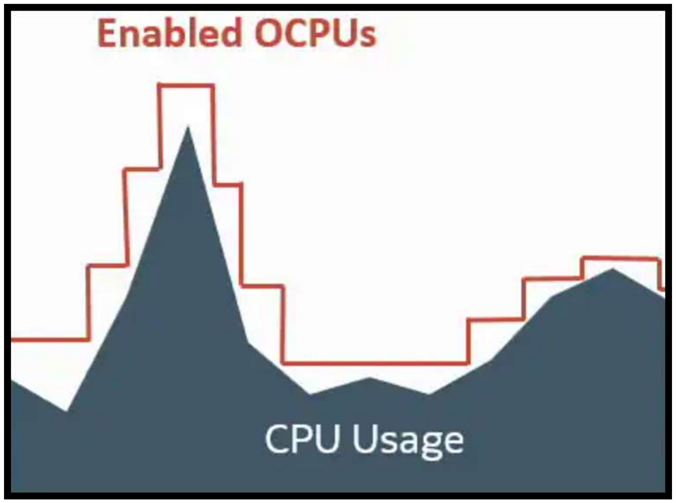
CPU SCALING

The first thing to understand is how Exadata Cloud Service charges for resources consumed by the system. The billing is based upon the number of cores allocated to the virtual machines in which the databases are running.

When your workload increases you must scale up the number of cores allocated to the virtual machine, and when workload goes down you must scale down the cores allocated to the virtual machine.

If you statically set the size of the VMs, you will most likely size them high to avoid having to frequently resize them as workload changes. Unfortunately, that also means you are probably paying for unnecessary capacity.

Exadata Cloud Service supports a rich set of APIs that allow you to scale your virtual machines up and down online, while the database is running, with no disruption to the database.



CPU SCALING

The criteria for performing a scale up or down operation is dependent upon a few different parameters and algorithms that you must set:

The resource you measure:
Typically, you measure CPU utilization within the VM. Since you are scaling CPU, this is the logical resource to measure.

Minimum resource threshold:
A point at which you declare the resource is low. For example, you may decide your CPU utilization should not go below 40%.

Maximum resource threshold:
A point at which you declare the resource is high. For example, you may decide your CPU utilization should not go above 60%.

The frequency of measurement:
How often you take a measurement and evaluate it against your criteria.

Interpretation of the measurement:
This algorithm determines how a measurement will be interpreted and when your script will initiate a scaling operation.

Delay for Scale Down Operations:
To provide optimum service levels, it may be best to not immediately scale down when load drops. This determines how long should you wait before initiating a scale-down operation.

CPU SCALING

- You can create a script using a scripting language of your choice.
- This script to run in the VMs you want to auto scale, or you can orchestrate the process from another server.
- We recommend you set up the script to automatically start and run as a daemon in the background. The script should basically loop indefinitely, measuring the resource (CPU utilization) every period and evaluating the interpreted results of the measurements against the thresholds. If a scaling operation is determined to be required, the script should call the API to scale the VM by the desired amount.
- If you don’t want to deal with writing and supporting your own script, Oracle RAC Pack has developed an autoscaling script for Exadata Cloud Service (on OCI or Exadata Cloud@Customer) that you are free to use. You can download it from MOS Note 2719916.1. It takes many of the parameters discussed above and implements some additional features.

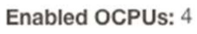
2. MANUAL SCALING

CPU SCALING

- VM Cluster has 4 OCPUs
- 2 OCPUs per node
- 2 cores with 2 threads per core = 4 vCPUs (or threads)

> Database parameter CPU_COUNT is 4

```
SQL > show parameter cpu_count

NAME      TYPE VALUE
____________________________________
cpu_count      integer 4
```

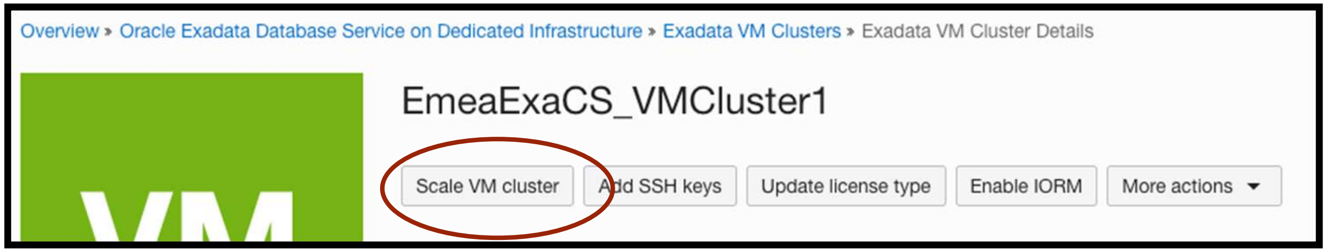


Enabled OCPUs: 4

- Scale VM Cluster up to 6 OCPUs
- 3 OCPUs per node
- 3 cores with 2 threads per core = 6 vCPUs (or threads)

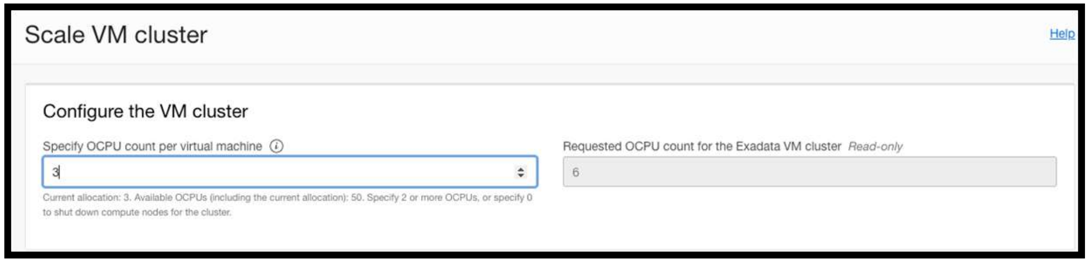
> Database parameter CPU_COUNT is 6

CPU SCALING



Overview » Oracle Exadata Database Service on Dedicated Infrastructure » Exadata VM Clusters » Exadata VM Cluster Details

EmeaExaCS_VMCluster1



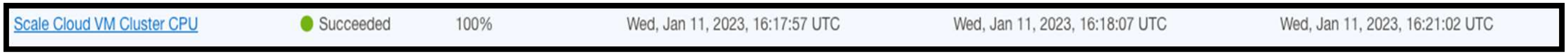
Scale VM cluster

Configure the VM cluster

Specify OCPU count per virtual machine: 3

Current allocation: 3. Available OCPUs (including the current allocation): 50. Specify 2 or more OCPUs, or specify 0 to shut down compute nodes for the cluster.

Requested OCPU count for the Exadata VM cluster *Read-only*: 6



Scale Cloud VM Cluster CPU	● Succeeded	100%	Wed, Jan 11, 2023, 16:17:57 UTC	Wed, Jan 11, 2023, 16:18:07 UTC	Wed, Jan 11, 2023, 16:21:02 UTC

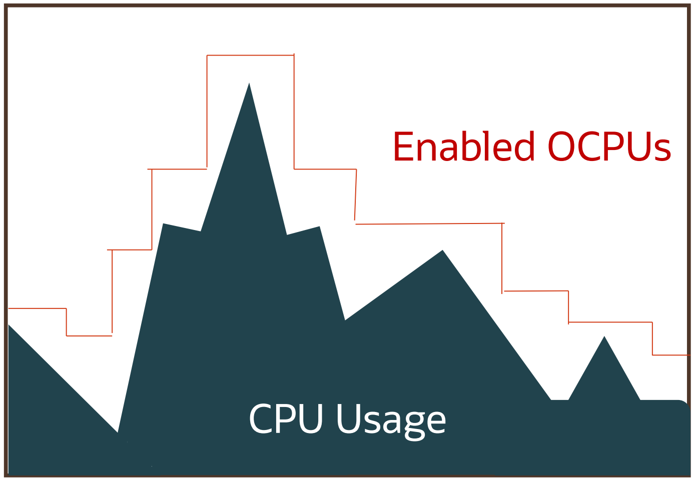
```
SQL > show parameter cpu_count

NAME       TYPE  VALUE
____________________________
cpu_count         integer 6
```

Enabled OCPUs: 6

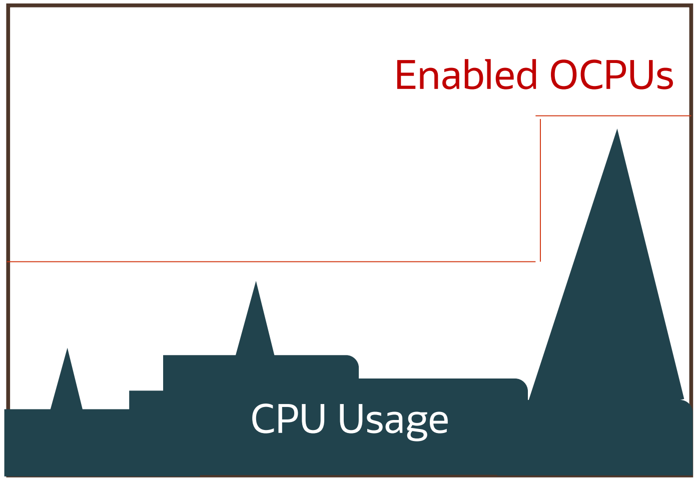
3. DYNAMIC SCALING

ODyS

ODyS (Oracle Dynamic Scaling) engine is a new tool to automate the OCI Database System (ExaCS/ExaC@C) scale-up or scale-down based on CPU load or scheduling.



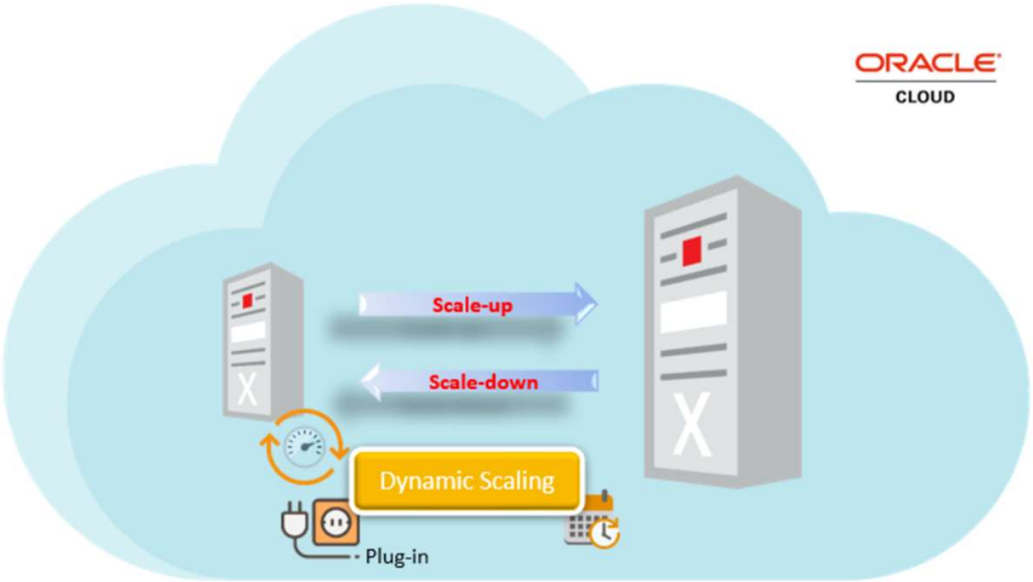
CPU load



Scheduling

ODyS

(ODyS) Oracle Dynamic Scaling engine - Scale-up and Scale-down automation utility for OCI DB System (ExaCS/ExaC@C) **(Doc ID 2719916.1)**

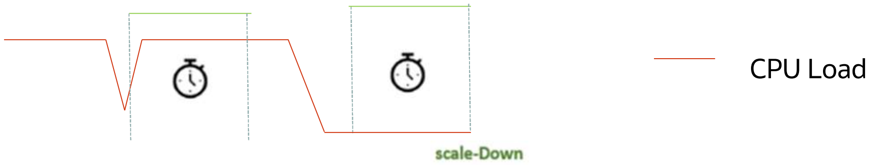


ODyS

- Oracle DynamicScaling can be executed as standalone executable or as daemon on one or more ExaCS compute nodes or ExaC@C vmcluster nodes.

- By default DynamicScaling is monitoring the CPUs with very limited host impact and if the load goes over the Maximum CPU threshold ("--maxthreshold") for an interval of time ("--interval"), it will automatically will scale-up the OCPU by a factor ("--ocpu") till a maximum limit ("--maxocpu").

- If the load goes under the Minimum CPU threshold ("--minthreshold") for an interval of time ("--interval") scale down will be executed util the minimum limit ("--minocpu") of ocpu.



- If a valid cluster filesystem (ACFS) is provided, DynamicScaling will consider the load of all nodes (where DynamicScaling is running) and it will perform a scale-UP/DOWN based on the **cluster node load (average/max)**.

ODyS

```
dynamicscaling.bin --db-system-id <DB system OCID>
                     | --cloud-vm-cluster-id <cloud VM cluster OCID>
                     | --vm-cluster-id <VM cluster OCID> --ociregion <DB System region>
                   --tenancy-id <tenancy OCID>
                   --user-id <user OCID>
                   --keyfingerprint <key finger print>
                   --privatekey <private key path>
                  [--cacert <cert file path>]
                  [--proxyhost <host> --proxyport <port> [--proxyid <user ID> --proxypass <password file>]]
                  [--maxocpu <Maximum OCPU Number>]
                  [--minocpu <Minimum OCPU Number>]
                  [--ocpu <Number of OCPU scale factor>] | [--ocpuup <scaleUp factor> --ocpudown <scaleDown factor>]
                  [--interval <Number of seconds>]
                  [--maxthreshold <maximum Load threshold>]
                  [--minthreshold <minimum Load threshold>]
                  [--acfs <writtable cluster filesystem path> [--loadtype <nodes load type: max or avg>]]
                  [--shape <system shape>]
                  [--nodecount <system node count>]
                  [--logfile <log file name>]
                  [--logpath <log file path>]
                  [--nolog]
                  [--nodaemon]
                  [--plugin <plugin path>]
                  [--scheduling <Dayname:hrmin-hrmax:cpu;>]
                  [--topic-id <notification topic OCID> | --mailx <comma separated recipient mail address>
                  [--sopnum <scale operations threshold>]]
                  [--dryrun]
```

ODyS

- Some of the parameters have default values that can be override.
- DynamicScaling is collecting OCI region, DB System maximum OCPU and scaling factor directly from the system.

```
dynamicscaling OPTIONS
  --db-system-id        Database system OCID
  --cloud-vm-cluster-id  Cloud VM cluster OCID (ExaCS systems)
  --vm-cluster-id       VM cluster OCID (ExaCC systems)
  --tenancy-id          Tenancy OCID
  --user-id             User OCID
  --keyfingerprint      User key Finger Print
  --privatekey          User private key path
  --cacert              Alternate CA public key
  --proxyHost           HTTP proxy server
  --proxyPort           HTTP proxy server port (Default: 80)
  --proxyId             HTTP proxy server username
  --proxyPass           HTTP proxy server password file
  --maxocpu             Max OCPU number (Default: DBSystem max OCPU)
  --minocpu             Min OCPU number (Default: 4)
  --ocpu                Number of OCPU scale factor (Default: DBSystem Number of Compute Nodes)
```

ODyS

--ocpuup	Number of OCPU scale factor for scale-up operations
--ocpudown	Number of OCPU scale factor for scale-down operations
--interval	**Number of seconds (Default: 180)**
--maxthreshold	**Maximum Load threshold 20-90 (Default: 80)**
--minthreshold	**Minimum Load threshold 10-80 (Default: 60)**
--acfs	**Writable cluster filesystem path for dynamicscaling cluster aware execution**
--loadtype	Nodes load type: max or avg (Default: max)
--shape	System Shape
--nodecount	System nodecount (ExaCC X8M/X9M)
--logfile	Log file name (Default: dynamicscaling.log)
--logpath	Log file path (Default: /tmp)
--nolog	It will not make the log '/tmp/dynamicscaling.log'
--nodaemon	**It will run as standlone**
--plugin	Plugin path, must return integer values 0-100
--scheduling	**Scheduling in the format : 'Dayname:hrmin-hrmax:cpu;Dayname:hrmin-hrmax:cpu;'**
--topic-id	Notification topic OCID
--mailx	Comma separated recipient mail address
--sopnum	Consecutive scale operations threshold number (Default: 5)
--dryrun	The scaling is not executed
--ociregion	OCI System Region

ODyS

Download rpm from note

```
dynamicscaling-2.0.1-23.el7.x86_64.rpm
```

Install

```
sudo rpm -i dynamicscaling-2.0.1-23.el7.x86_64.rpm
warning: dynamicscaling-2.0.1-23.el7.x86_64.rpm: Header V4 RSA/SHA1 Signature, key ID 46caeffa: NOKEY
dynamicscaling-2.0.1.23 binary has been installed on /opt/dynamicscaling succesfully!
```

The following files are created under '/opt/dynamicscaling':

```
/opt/dynamicscaling/
└── dynamicscaling.bin
0 directories, 1 fil
```

/opt/dynamicscaling/dynamicscaling.bin status

```
|░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░|
 Dynamicscaling for ExaCC & ExaCS - Version: 2.0.1-23
 Copyright (c) 2020-2022 Oracle and/or its affiliates.
----------------------------------------------------------------
 Author: Ruggero Citton <ruggero.citton@oracle.com>
 RAC Pack, Cloud Innovation and Solution Engineering Team
|░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░|

Dynamicscaling is not running
```

ODyS

```
/opt/dynamicscaling/dynamicscaling.bin getocpu \
--cloud-vm-cluster-id ocid1.cloudvmcluster. xxxxxx  \
--tenancy-id ocid1.tenancy.oc1.. xxxxxx \
--user-id ocid1.user.oc1..xxxxxx \
--keyfingerprint xxxxxx \
--privatekey /home/opc/.oci/API-Private_Key.pem
```

```
░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░|
 Dynamicscaling for ExaCC & ExaCS - Version: 2.0.1-19
 Copyright (c) 2020-2022 Oracle and/or its affiliates.
------------------------------------------------------------
 Author: xxxxxxxxxxxxxxxxxxxxx
 RAC Pack, Cloud Innovation and Solution Engineering Team
|░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░|

INFO: 2023-01-11 17:16:05: Checking DB System status

- ----------------------------------------------------------
  Cloud VM cluster OCID  : ocid1.cloudvmcluster. xxxxxx
  Tenancy OCID          : ocid1.tenancy.oc1.. xxxxxx
  User    OCID          : ocid1.user.oc1..xxxxxx
  Key finger print      : xxxxxx
  Private key path      : /home/opc/.oci/API-Private_Key.pem
- ----------------------------------------------------------
  DB System status      : AVAILABLE
  Current OCPU          : 4
  Current physical CPUs  : 1
- ----------------------------------------------------------
  Dynamicscaling Log file: '/tmp/dynamicscaling.log'
------------------------------------------------------------
```

ODyS

```
/opt/dynamicscaling/dynamicscaling.bin setocp \
--cloud-vm-cluster-id ocid1.cloudvmcluster. xxxxxx  \
--tenancy-id ocid1.tenancy.oc1.. xxxxxx \
--user-id ocid1.user.oc1..xxxxxx \
--keyfingerprint xxxxxx \
--privatekey /home/opc/.oci/API-Private_Key.pem
--ocpu 6
```

```
|░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░|
 Dynamicscaling for ExaCC & ExaCS - Version: 2.0.1-23
 Copyright (c) 2020-2022 Oracle and/or its affiliates.
------------------------------------------------------------
 Author: Ruggero Citton <ruggero.citton@oracle.com>
 RAC Pack, Cloud Innovation and Solution Engineering Team
|░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░░|


INFO: 2023-01-12 10:22:00: dynamicscaling log file at '/tmp/dynamicscaling.log'
INFO: 2023-01-12 10:22:01: Checking DB System status
INFO: 2023-01-12 10:22:02: Current DB System status=AVAILABLE
INFO: 2023-01-12 10:22:02: Checking current core count
INFO: 2023-01-12 10:22:02: Current OCPU=4
SUCCESS: 2023-01-12 10:22:03: Scaling-up to OCPU=6 in progress, please wait...
```

ODyS

```
/opt/dynamicscaling/dynamicscaling.bin **getocpu** \
--cloud-vm-cluster-id ocid1.cloudvmcluster. xxxxxx  \
--tenancy-id ocid1.tenancy.oc1.. xxxxxx \
--user-id ocid1.user.oc1..xxxxxx \
--keyfingerprint xxxxxx \
--privatekey /home/opc/.oci/API-Private_Key.pem
```

```
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒|
 Dynamicscaling for ExaCC & ExaCS - Version: 2.0.1-19
 Copyright (c) 2020-2022 Oracle and/or its affiliates.
-----------------------------------------------------------
 Author: Ruggero Citton <ruggero.citton@oracle.com>
 RAC Pack, Cloud Innovation and Solution Engineering Team
|▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒|

INFO: 2023-01-11 17:16:05: Checking DB System status

- ---------------------------------------------------------
  Cloud VM cluster OCID  : ocid1.cloudvmcluster. xxxxxx
  Tenancy OCID          : ocid1.tenancy.oc1.. xxxxxx
  User    OCID          : ocid1.user.oc1..xxxxxx
  Key finger print      : xxxxxx
  Private key path      : /home/opc/.oci/API-Private_Key.pem
- ---------------------------------------------------------
  DB System status      : AVAILABLE
  Current OCPU          : 6
  Current physical CPUs  : 1
- ---------------------------------------------------------
  Dynamicscaling Log file: '/tmp/dynamicscaling.log'
-----------------------------------------------------------
```

ODyS

```
/opt/dynamicscaling/dynamicscaling.bin \
--nodaemon \
--cloud-vm-cluster-id ocid1.cloudvmcluster.xxxxxx \
--tenancy-id ocid1.tenancy.oc1.. xxxxxx \
--user-id ocid1.user.oc1.. xxxxxx \
--keyfingerprint xxxxxx  \
--privatekey /home/opc/.oci/API-Private_Key.pem \
--interval 300 \
--maxthreshold 80 \
--minthreshold 60 \
--maxocpu 6 \
--minocpu 4 \
--ocpu 2 \
--plugin '/home/opc/DynamicScaling/dynamicscaling_LoadAverage_plugin.sh'
```

DynamicScaling can be executed as standalone executable or as a daemon. As a best practice execute Dynamicscale as standalone ("--nodaemon") to check for any errors you may get on your environment.

If everything is working as expected you can execute DynamicScaling as a daemon and check the trace created as "***/tmp/dynamicscaling.log***" for any errors.

ODyS

Oracle DynamicScaling 'Load Average' plugin

It will use the load average percentage. The load average time is driven by 'LOAD_AVG_TIME' set to 1 minute

```
#!/bin/bash# ---------------------------------------------------------------------------
# Copyright (c) 2020, 2021 Oracle and/or its affiliates. All rights reserved.
#
# DynamicScaling 'Load Average' plugin example
# File_name: dynamicscaling_LoadAverage_plugin.sh
#
# Author: xxxxx
# RAC Pack, Cloud Innovation and Solution Engineering Team
# ---------------------------------------------------------------------------

# Load average time to consider

LOAD_AVG_TIME=1M
usage() {
 echo
 echo "Usage: 'basename $0' 1M|5M|15M"
}
main() {
 if [ $# -eq 0 ]; then
  usage
  exit
 fi
 for cmd_operation in $*
 do

  case $1 in

   1M) LOAD_AVERAGE=$(awk '{avg_1m=($1)}  END {printf "%3.2f", avg_1m}' /proc/loadavg)
     ;;

   5M) LOAD_AVERAGE=$(awk '{avg_5m=($2)}  END {printf "%3.2f", avg_5m}' /proc/loadavg)
     ;;

   15M) LOAD_AVERAGE=$(awk '{avg_15m=($3)} END {printf "%3.2f", avg_15m}' /proc/loadavg)
     ;;

    *) if [ "$1" != "" ]; then
      usage
      echo "Invalid argument: $1"
      exit
     fi
     ;;
   esac
 done

 local MAX=$(nproc --all)

 if [ "$MAX" -eq 0 ]; then
  local PERCENT=100

 else
  local PERCENT=$(bc <<< "$LOAD_AVERAGE*100/$MAX")
  if [ "$PERCENT" -gt 100 ]; then
   local PERCENT=100
  fi
 fi
 echo $PERCENT
}
main $LOAD_AVG_TIME
# ----------------------------------------------
# EndOfFile
# ----------------------------------------------
```

ODyS OS Service

How to run DynamicScaling at boot time (OS service)

Create a new systemd service unit. Create a new service unit file at "*/etc/systemd/system/dynamicscaling.service*" with the content below. (the service will execute as '***opc***' user)

```
[Unit]
Description=Dynamicscaling
Wants=network-online.target local-fs.target
After=network-online.target local-fs.target

[Service]
User=opc
Type=simple
Environment="PATH=/sbin:/bin:/usr/sbin:/usr/bin:/opt/oci-client/bin"
ExecStart=/bin/sh -c "/opt/dynamicscaling/dynamicscaling.bin <command options>"
TimeoutStartSec=300
PIDFile=/tmp/.dynamicscaling.pid
Restart=on-failure
RestartSec=5s
ExecStop=/bin/kill -s SIGINT $MAINPID

[Install]
WantedBy=multi-user.target
```

How to run DynamicScaling at boot time (OS service)

```
[Unit]
Description=Dynamicscaling - Scale-up and Scale-down automation utility for OCI DB System
Wants=network-online.target local-fs.target
After=network-online.target local-fs.target

[Service]
User=opc
Type=simple
Environment="PATH=/sbin:/bin:/usr/sbin:/usr/bin:/opt/oci-client/bin"
ExecStart=/bin/sh -c "/opt/dynamicscaling/dynamicscaling.bin --cloud-vm-cluster-id ocid1.cloudvmcluster.oc1.xxxxx --tenancy-id ocid1.tenancy.oc1..xxx --user-id ocid1.user.oc1..xxx --keyfingerprint xxx --privatekey /home/opc/.oci/API-Private_Key.pem --interval 300 --maxthreshold 80 --minthreshold 60 --maxocpu 6 --minocpu 4 --ocpu 2 --acfs /acfs01/odys"
TimeoutStartSec=300
PIDFile=/tmp/.dynamicscaling.pid
Restart=on-failure
RestartSec=5s

ExecStop=/bin/kill -s SIGINT $MAINPID

[Install]
WantedBy=multi-user.target
```

Note: before making the service run the same command (ExecStart) with "**--nodaemon**" to check for potential errors

How to run DynamicScaling at boot time (OS service)

Enable the systemd service unit

Reload the systemd process to consider newly created sample.service OR every time when sample.service gets modified.

```
# systemctl daemon-reload
```

Enable this service to start after reboot automatically.

```
# systemctl enable dynamicscaling.service
```

Start the service.

```
# systemctl start dynamicscaling.service
```

Check the Service Status

```
# systemctl status dynamicscaling.service

/opt/dynamicscaling/dynamicscaling.bin status

tail -f /tmp/dynamicscaling.log
```

Stop the Service Status

```
# systemctl stop dynamicscaling.service
```

ODyS

Considerations & Best Practices

ODyS Considerations & Best Practices

- EXACS required OCI IAM privileges are CLOUD_VM_CLUSTER_UPDATE and CLOUD_EXADATA_INFRASTRUCTURE_UPDATE for which required policy definition could be.

```
     Allow group <groupname> to use cloud-exadata-infrastructures IN TENANCY
     Allow group <groupname> to use cloud-vmclusters IN TENANCY
```

- EXAC@C required OCI IAM privileges are VM_CLUSTER_UPDATE and EXADATA_INFRASTRUCTURE_UPDATE for which required policy definition could be.

```
     Allow group <groupname> to use exadata-infrastructures IN TENANCY
     Allow group <groupname> to use vmclusters IN TENANCY
```

- DynamicScaling is making the log file as "*/tmp/dynamicscaling.log*", you should check the size of it and in case you can remove it.

- Once dynamicscaling is firing a scale up/down request, DB System needs about 3 minutes to complete the operation.

- Consider disabling dynamicscaling during your DB host lifesycle operations like patching, upgrade etc.

ODyS Considerations & Best Practices

- DynamicScaling allows you to run with minimum capacity in weekends. So make sure when you really want to scale up in those scheduled timeframe, you need to stop the running DynamicScaling on all nodes, and then go for manual DynamicScaling setocpu. If this is not happens, manual scaling still scale ups to desired value but again it scale down to configured values in scheduled DynamicScaling (scheduling is having priority over the load).

- Check all Possible Configurations (Note ID : 2834931.1)

- In case your load across the nodes is not balanced, review this note '***(ODyS) Oracle Dynamic Scaling - Remote Plug-in***' (see Note:2770544.1).

ODyS CONCLUSIONS

ODyS Conclusions

- The easiest way to reduce your TCO in the cloud is to stop paying for excess capacity when it is not required.

- Autoscaling is a great way to match workload demands with Exadata capacity, providing you true pay-for-use pricing. It is very simple using the published APIs to implement script-driven automation, tuned specifically for your workload.

- Try it out with your favourite scripting language (check this article) , or download a fully tested implementation from MyOracle Support. MOS Note 2719916.1. **(ODyS) Oracle Dynamic Scaling engine - Scale-up and Scale-down automation utility for OCI DB System (ExaCS/ExaC@C) (Doc ID 2719916.1)**

THANKS