

**Autoscaling - Scale-up and Scale-down automation utility for OCI DB System (ExaCS/ExaCC) (Doc ID 2719916.1)****In this Document**[Abstract](#)[History](#)[Details](#)[How Autoscaling Works](#)[Autoscaling Default Parameters](#)[Autoscaling Load Plugin Usage](#)[Example 1 - Autoscaling 'CPU\\_USAGE' plugin](#)[Example 2 - Autoscaling 'Load Average' plugin](#)[Autoscaling Requirement](#)[Autoscaling Scheduling](#)[How to Install Autoscaling](#)[Autoscaling Usage](#)[How to run Autoscaling at boot time](#)[How to stop Autoscaling](#)[Autoscaling Additional Functions](#)[Autoscaling Status](#)[Autoscaling GetOCPU](#)[Autoscaling SetOCPU](#)[Autoscaling Considerations](#)[Autoscaling Know Problem](#)[References](#)

Oracle Confidential INTERNAL - Do not distribute to customer (OracleConfidential).  
Reason: See note in History [This section is not visible to customers.]

**APPLIES TO:**

Oracle Database Cloud Exadata Service  
Gen 2 Exadata Cloud at Customer  
Linux x86-64  
Linux x86-64 on Oracle Public Cloud

**ABSTRACT**

Autoscaling is a new tool to automate the OCI Database System (ExaCS/ExaCC) scale-up or scale-down based on CPU load or scheduling.

**HISTORY**

Author: Ruggero Citton - RAC Pack, Cloud Innovation and Solution Engineering Team

Create Date : October- 2020

Update Date: February- 2021

Latest Autoscale:

[Autoscaling for Linux el7](#) Version: 20210202 - \$Revision: 2.0.1.15 \$ (### The Autoscaling RPM is NOT available here ###)

Autoscaling has been tested and adopted by some key ExaCS/ExaCC customers with excellent results. Autoscaling is an interim solution waiting on Control Plane integration for which there is not ETA yet.

If you (or your customers) wish to have access to Autoscaling, please contact ExaCS & ExaCC Product manager:  
[mathew.steinberg@oracle.com](mailto:mathew.steinberg@oracle.com), [bob.thome@oracle.com](mailto:bob.thome@oracle.com) (keep in CC development management : [sanjay.singh@oracle.com](mailto:sanjay.singh@oracle.com))

for an exception approval. Autoscaling is not public due to Product manager's request.

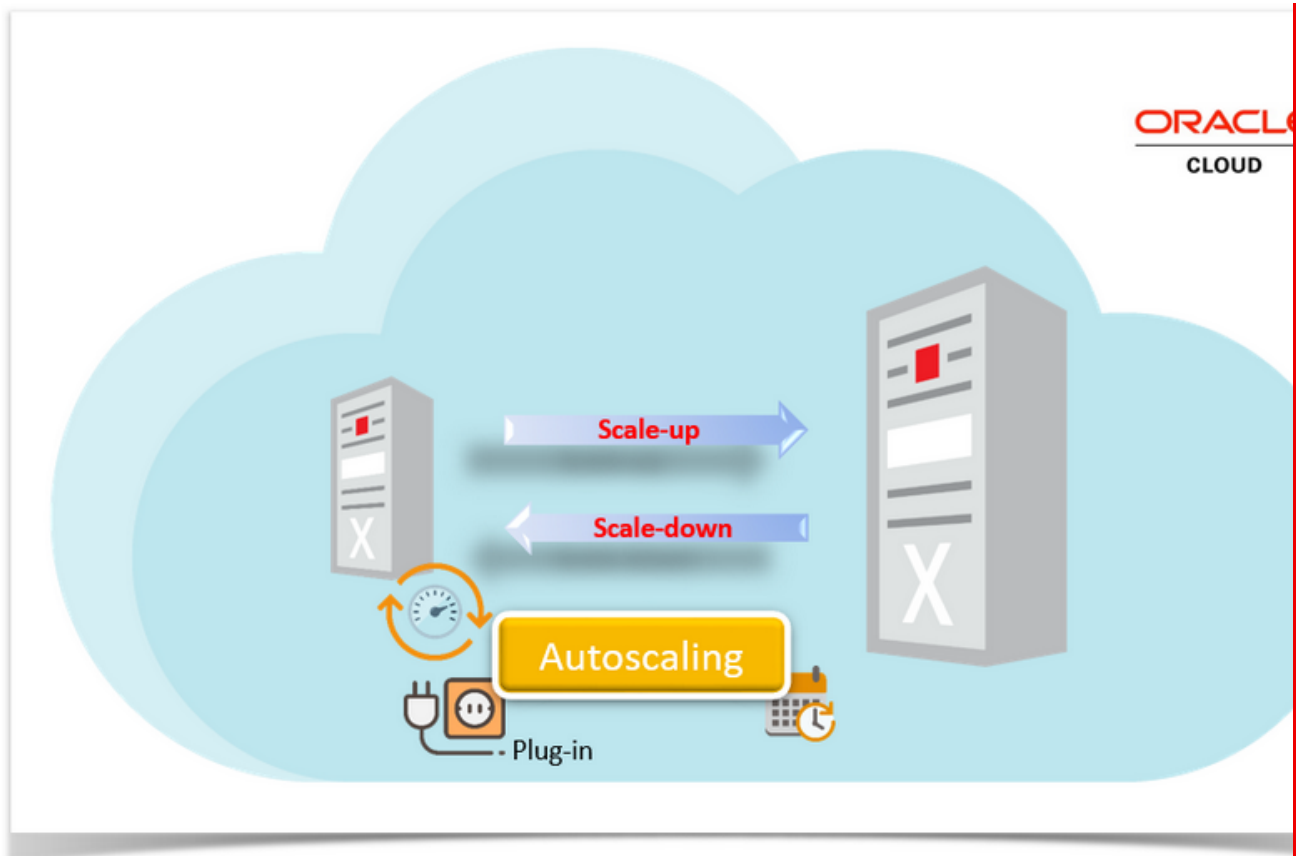
[This section is not visible to customers.]

DETAILS

```
### |
### -----
### Autoscaling version 2.0.1
###
### Copyright (c) 2020-2021 Oracle and/or its affiliates. All rights reserved.
### -----
### Disclaimer:
###
### EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE INFORMATION, SOFTWARE,
### PROVIDED ON AN \"AS IS\" AND \"AS AVAILABLE\" BASIS. ORACLE EXPRESSLY DISCLAIMS
### ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT
### LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
### PURPOSE AND NON-INFRINGEMENT. ORACLE MAKES NO WARRANTY THAT: (A) THE RESULTS
### THAT MAY BE OBTAINED FROM THE USE OF THE SOFTWARE WILL BE ACCURATE OR
### RELIABLE; OR (B) THE INFORMATION, OR OTHER MATERIAL OBTAINED WILL MEET YOUR
### EXPECTATIONS. ANY CONTENT, MATERIALS, INFORMATION OR SOFTWARE DOWNLOADED OR
### OTHERWISE OBTAINED IS DONE AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE
### NO RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
### RESULTS FROM THE DOWNLOAD OF ANY CONTENT, MATERIALS, INFORMATION OR SOFTWARE.
###
### ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO THE SOFTWARE AT ANY
### TIME WITHOUT NOTICE.
###
### Limitation of Liability:
###
### IN NO EVENT SHALL ORACLE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
### SPECIAL OR CONSEQUENTIAL DAMAGES, OR DAMAGES FOR LOSS OF PROFITS, REVENUE,
### DATA OR USE, INCURRED BY YOU OR ANY THIRD PARTY, WHETHER IN AN ACTION IN
### CONTRACT OR TORT, ARISING FROM YOUR ACCESS TO, OR USE OF, THE SOFTWARE.
### -----
### The tool has been tested and appears to work as intended.
### -----
### |
```

To fill a bug against AUTOSCALING : Product ID=5, Component=OPSM, Subcomponent=AUTOSCALING[This section is not visible to customers.]

When auto scaling is running, if your workload requires additional CPUs the database system automatically will uses the resources without any manual intervention required.



### How Autoscaling Works

Autoscaling can be executed as standalone executable or as daemon on one or more ExaCS compute nodes or ExaCC vmcluster nodes. By default Autoscaling 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 until the minimum limit ("--minocpu") of ocpu. If a valid cluster filesystem (ACFS) is provided, Autoscaling will consider the load of all nodes (where autoscaling is running) and it will perform a scale-UP/DOWN based on the **cluster node load average**.

```
autoscaling.bin --db-system-id <DB system OCID>
                | --cloud-vm-cluster-id <cloud VM cluster OCID>
                | --vm-cluster-id <VM cluster OCID> --oci-region <DB System region>
--tenancy-id <tenancy OCID>
--user-id <user OCID>
--keyfingerprint <key finger print>
--privatekey <private key path>
[--proxyhost <host> -proxyport <port> [--proxyid <user ID> --proxypass <password>]]
[--maxocpu <Maximum OCPU Number>]
[--minocpu <Minimum OCPU Number>]
[--ocpu <Number of OCPU scale factor>]
[--interval <Number of seconds>]
[--maxthreshold <maximum Load threshold>]
[--minthreshold <minimum Load threshold>]
[--acfs <writable cluster filesystem path>]
[--nolog]
[--nodaemon]
[--plugin <plugin path>]
[--scheduling <Dayname:hrmin-hrmax:cpu>]
[--dryrun]
```

#### autoscaling 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
--proxyHost	HTTP proxy server
--proxyPort	HTTP proxy server port (Default: 80)
--proxyId	HTTP proxy server username
--proxyPass	HTTP proxy server password
--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)
--interval	Number of seconds (Default: 180)
--maxthreshold	Maximum Load threshold 0-100 (Default: 80)

```
--minthreshold      Minimum Load threshold 0-100 (Default: 60)
--acfs              Writable cluster filesystem path for autoscaling cluster aware execution
--nolog            It will not make the log '/tmp/autoscaling.log'
--nodaemon          It will run as standalone
--plugin            Plugin path, must return integer values 0-100
--scheduling         Scheduling in the format : 'Dayname:hrmin-hrmax:cpu;Dayname:hrmin-hrmax:cpu;'
--dryrun            The scaling is not executed
--ociregion         OCI System Region
```

Using OCI Client ([oci-cli](#)):

```
autoscaling.bin --ocicli
--db-system-id <DB system OCID>
|--cloud-vm-cluster-id <cloud VM cluster OCID>
|--vm-cluster-id <VM cluster OCID>
[--ociprofile <oci profile name>]
[--maxocpu <Maximum OCPU Number>]
[--minocpu <Minimum OCPU Number>]
[--ocpu <Number of OCPU scale factor>]
[--interval <Number of seconds>]
[--maxthreshold <Maximum Load threshold>]
[--minthreshold <Minimum Load threshold>]
[--acfs <writable cluster filesystem path>]
[--nolog]
[--nodaemon]
[--scheduling <Dayname:hrmin-hrmax:cpu;>]
[--dryrun]

autoscaling.bin OPTIONS
--ocicli          OCI client usage
--db-system-id    Database system OCID
--cloud-vm-cluster-id  Cloud VM cluster OCID (ExaCS systems)
--vm-cluster-id   VM cluster OCID (ExaCC systems)
--ociprofile      OCI profile name from $HOME/.oci/config (Default: 'DEFAULT')
--maxocpu         Maximum OCPU number (Default: DBSystem max OCPU)
--minocpu         Minimum OCPU number (Default: 4)
--ocpu            Number of OCPU scale factor (Default: DBSystem Number of Compute Nodes)
--interval        Number of seconds (Default: 180)
--maxthreshold    Maximum Load threshold 0-100 (Default: 80)
--minthreshold    Minimum Load threshold 0-100 (Default: 60)
--acfs            Writable cluster filesystem path for autoscaling cluster aware execution
--nolog           It will not make the log '/tmp/autoscaling.log'
--plugin          Plugin path, must return integer values 0-100
--nodaemon        It will run as standalone
--scheduling       Scheduling in the format : 'Dayname:hrmin-hrmax:cpu;Dayname:hrmin-hrmax:cpu;'
--dryrun          The scaling is not executed
```

### Autoscaling Default Parameters

Autoscaling is having many default parameters you could override with the related parameter usage

- Default OCI client profile name ("--ociprofile") : "DEFAULT"
- Default Upper OCPU limit ("--maxocpu") : DBSystem max OCPU
- Default Lower OCPU limit ("--minocpu") : 4
- Default number of OCPU scale factor ("--ocpu") : DBSystem Number of Compute Nodes
- Default number of seconds between CPU load measure ("--interval"): 180 secs
- Default Maximum load threshold : 80%
- Default Minimum load threshold : 60%
- Default Proxy port: 80

Autoscaling is collecting OCI region, DB System maximum OCPU and scaling factor directly from the system.

### Autoscaling Load Plugin Usage

By default Autoscaling measures comprehensive CPU load to take actions. Users can provide their own load plugin, it could be whatever executable (bash, perl, c, c++, python, etc. executable) that returns an integer value 0-100. Autoscaling will call the plugin ("--plugin") and will perform the job.

#### Example 1 - Autoscaling 'CPU\_USAGE' plugin

It will use simple CPU usage

```
#!/bin/bash

# -----#
# Autoscaling 'CPU_USAGE' plugin example
# File_name: autoscaling_CPU_USAGE_plugin.sh
#
# Author: Ruggero Citton -
# RAC Pack, Cloud Innovation and Solution Engineering Team
# -----#

# The 1st iteration of top in batch mode (-b) returns the percentages since boot, then we need
```

```
# at least two iterations (-n 2) to get the current percentage. To speed up the measure
# we set the delay (-d 0.01).
# Top return CPU usage as user, system processes and nice processes, we sum all three.
CPU_USAGE=$(top -bn 2 -d 0.01 | grep '^%Cpu' | tail -n 1 | gawk '{print $2+$4+$6}' | cut -d. -f1)
echo $CPU_USAGE

# -----
# EndOfFile
# -----
```

### Example 2 - Autoscaling '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.
#
# Autoscaling 'Load Average' plugin example
# File_name: autoscaling_LoadAverage_plugin.sh
#
# Author: Ruggero Citton -
# 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=$(nproc --all)
    if [ "$SMAX" -eq 0 ]; then
        local PERCENT=100
    else
        local PERCENT=$((bc <<< "$LOAD_AVERAGE*100/$MAX"))
    fi
    echo $PERCENT
}

main $LOAD_AVG_TIME

# -----
# EndOfFile
# -----
```

Usage example:

```
./autoscaling.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 <user keyfingerprint> \
    --privatekey <user privatekey> \
    --interval 300 \
    --maxthreshold 80 \
    --minthreshold 60 \
    --maxocpu 96 \
    --minocpu 22 \
    --ociregion eu-frankfurt-1 \
```

```
--plugin '<path>/autoscaling_LoadAverage_plugin.sh'
```

### Autoscaling Requirement

Autoscaling may leverage on OCI Client ([oci-cli](#)). Once installed and configured, you can specify the oci-cli profile to be used with "--ociprofile" Autoscaling command option if different from the default.

If OCI Client is not in use additional parameters are required:

```
--tenancy-id      Tenancy OCID
--user-id         User OCID
--keyfingerprint  User key Finger Print
--privatekey      User private key path
```

see "[Required Keys and OCIDs](#)" (generate the key with no passphrase)

### Autoscaling Scheduling

Autoscaling considers the scheduling with priority over the load measure. If the scheduling option is in use and if it's applicable it will use the scheduling setting. If the scheduling option is in use and if it's not applicable due to time, the load will be considered.

Expected scheduling form is as follows:

```
Dayname:hrmin-hrmax:cpu;Dayname:hrmin-hrmax:cpu;
where Dayname = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
      hrmin = integer between 0 to 23
      hrmax = integer between 0 to 23
      cpu = integer > 0 and multiple of 2
```

An example of scheduling option is as follows:

```
"Monday:0-8:2;Monday:9-18:26;Tuesday:0-23:12;Wednesday:0-8:2;Wednesday:8-16:26;Wednesday:16-23:22;Saturday:0-23:2;Sunday:0-23:2"
```

Following above example,

```
on Monday      between 0am and 8.59am --> 2 OCPU will be in use
on Monday      between 9am and 6.59pm --> 26 OCPU will be in use
.
on Tuesday     between 0am and 11.59pm --> 12 OCPU will be in use
.
on Wednesday   between 0am and 8.59am --> 2 OCPU will be in use
on Wednesday   between 8am and 4.59pm --> 26 OCPU will be in use
on Wednesday   between 4pm and 11.59pm --> 22 OCPU will be in use
.
on Saturday & Sunday only 2 OCPU will be in use
on other days/time the load will be considered instead
```

### How to Install Autoscaling

autoscale can be installed using the RPM (RedHat Package Manager) command as following:

```
# rpm -i Autoscaling-2.0.1-X.e17.x86_64.rpm
(*) X=version number
```

or updating an installed version, issuing:

```
# rpm -Uvh Autoscaling-2.0.1-X.e17.x86_64.rpm
(*) X=version number
```

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

```
/opt/autoscaling/
└─ autoscaling.bin
0 directories, 1 file
```

### Autoscaling Usage

Autoscaling can be executed as standalone executable or as a daemon. As a best practice execute autoscale as standalone ("--nodaemon") to check for any errors you may get on your environment. If everything is working as expected you can execute autoscaling as a daemon. You could also start Autoscaling as a daemon and check the trace created under *"/tmp/autoscaling.log"* for any errors. You may avoid any log creation using *"--nolog"* (not recommended)

### How to run Autoscaling at boot time

1. Create a new systemd service unit

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

```
[Unit]
Description=Autoscaling
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/autoscaling/autoscaling.bin <command options>"

TimeoutStartSec=300
PIDFile=/tmp/.autoscaling.pid
Restart=on-failure
RestartSec=5s

ExecStop=/bin/kill -s SIGINT $MAINPID

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

example (using DB System API "--db-system-id", if your env has been migrated to the new API you need specify "--cloud-vm-cluster-id" or on ExaCC "--vm-cluster-id") :

```
# Copyright (c) 2020, 2021 Oracle and/or its affiliates. All rights reserved.
#
# Oracle autoscaling.service

[Unit]
Description=Autoscaling - 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/autoscaling/autoscaling.bin --db-system-id <dbsystem.oid> --tenancy-id <tenancy.oid> --user-id <user.oid> --keyfingerprint <...> --privatekey <...> --interval 300 --maxthreshold 80 --minthreshold 60 --maxocpu 96 --minocpu 68 --ocpu 4 --acfs /acfs01/.autoscaling --scheduling 'Saturday:0-23:28;Sunday:0-23:28'"

TimeoutStartSec=300
PIDFile=/tmp/.autoscaling.pid
Restart=on-failure
RestartSec=5s

ExecStop=/bin/kill -s SIGINT $MAINPID

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

Note1: Option "--nodaemon" can not be used on systemd service unit  
 Note2: Service definition is using 'User=opc', autoscaling will execute as opc user  
 Note3: ACFS:'/acfs01/.autoscaling' need to be writable by opc user

## 2. Enable the systemd service unit

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

```
# systemctl daemon-reload
```

b. Enable this service to start after reboot automatically.

```
# systemctl enable autoscaling.service
```

c. Start the service.

```
# systemctl start autoscaling.service
```

d. Check the Service Status

```
# systemctl status autoscaling.service
```

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

## How to stop Autoscaling

Autoscaling can be stopped by issuing:

```
# autoscaling.bin stop|shutdown|alt|kill
```

or if running as OS service

```
# systemctl stop autoscaling.service
```

## Autoscaling Additional Functions

### Autoscaling Status

Using the "status" function

```
# autoscaling.bin status
```

you can get information about the autoscaling by specifying the status option, i.e.:

```
# ./autoscaling.bin status

-----
- Autoscaling is running
Process ID : 336635
Start Date : Mon Nov 23 08:02:48 2020
Running command: autoscaling.bin \
    --db-system-id <DB system OCID> |--cloud-vm-cluster-id <cloud VM cluster OCID> |--vm-cluster-id <VM cluster
OCID> \
    --tenancy-id <tenancy.oid> \
    --user-id <user.oid> \
    --keyfingerprint <.....> \
    --privatekey <.....> \
    --interval 300 \
    --maxthreshold 80 \
    --minthreshold 60 \
    --maxocpu 96 \
    --minocpu 68 \
    --ocpu 4 \
    --acfs /acfs01 \
    --scheduling Saturday:0-23:28;Sunday:0-23:28
-----
Autotrace Log file: '/tmp/autoscaling.log'
-----
```

### Autoscaling GetOCPU

Using the "getocpu" function you can get the current OCPU count:

```
Usage:
autoscaling.bin getocpu --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>
    [--proxyhost <host> -proxyport <port> [--proxyid <user ID> --proxypass <password>]]

autoscaling.bin getocpu 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
--ociregion          OCI System Region
--proxyHost          HTTP proxy server
--proxyPort          HTTP proxy server port (Default: 80)
--proxyId            HTTP proxy server username
--proxyPass          HTTP proxy server password
```

Using OCI Client ([oci-cli](#)):

```
Usage:
autoscaling.bin getocpu --ocicli
    --db-system-id <DB system OCID> |--cloud-vm-cluster-id <cloud VM cluster OCID> |--vm-cluster-id <VM
cluster OCID>
    [--ociprofile <oci profile name>]

autoscaling.bin getocpu OPTIONS
--ocicli            OCI client usage
--db-system-id      Database system OCID
--cloud-vm-cluster-id Cloud VM cluster OCID (ExaCS systems)
--vm-cluster-id     VM cluster OCID (ExaCC systems)
--ociprofile         OCI profile name from '$HOME/.oci/config' (Default: 'DEFAULT')
```



## Autoscaling SetOCPU

Using the "setocpu" function you can set the OCPU:

```
Usage:
autoscaling.bin setocpu --db-system-id <DB system OCID> [--cloud-vm-cluster-id <cloud VM cluster OCID> |--vm-cluster-id <VM
cluster OCID> --oci-region <DB System region>
                        --tenancy-id <tenancy OCID>
                        --user-id <user OCID>
                        --keyfingerprint <key finger print>
                        --privatekey <private key path>
                        --ocpu <OCPU Number>
                        [--proxyhost <host> -proxyport <port> [--proxyid <user ID> --proxypass <password>]]

autoscaling.bin setocpu 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
--oci-region        OCI System Region
--proxyHost         HTTP proxy server
--proxyPort         HTTP proxy server port (Default: 80)
--proxyId           HTTP proxy server username
--proxyPass         HTTP proxy server password
--ocpu              OCPU number
```

Using OCI Client ([oci-cli](#)):

```
Usage:
autoscaling.bin setocpu --ocicli
                        --db-system-id <DB system OCID> |--cloud-vm-cluster-id <cloud VM cluster OCID> |--vm-cluster-id <VM
cluster OCID>
                        --ocpu <OCPU Number>
                        [--ociprofile <oci profile name>]

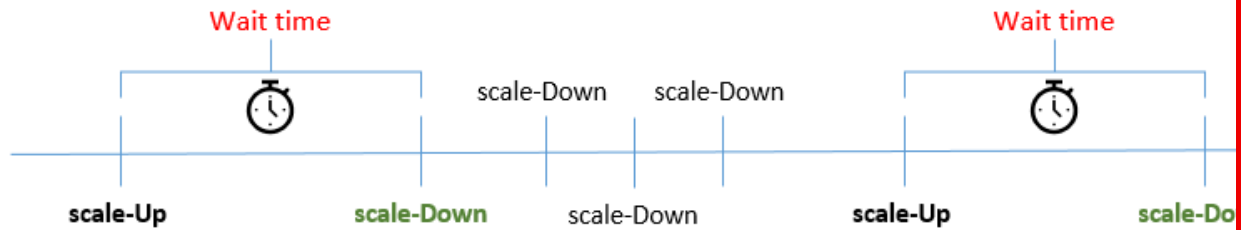
autoscaling.bin setocpu OPTIONS
--ocicli            OCI client usage
--db-system-id      Database system OCID
--cloud-vm-cluster-id Cloud VM cluster OCID (ExaCS systems)
--vm-cluster-id     VM cluster OCID (ExaCC systems)
--ociprofile        OCI profile name from '$HOME/.oci/config' (Default: 'DEFAULT')
--ocpu              OCPU number
```

## Autoscaling Considerations

- Autoscaling supports the '**Database System API**' ("--db-system-id") and the new '**Cluster VM API**' (on ExaCS "--cloud-vm-cluster-id" or on ExaCC "--vm-cluster-id"). If when running Autoscaling you get a warning about "*DB System status 'MIGRATED'*" you need to specify the "vm cluster id" instead.
- Autoscaling can be executed as '**opc**' user. If you start autoscaling using 'root' pay attention to "/tmp/autoscaling.log" permissions, to see if you need to start autoscaling with a different user.
- Autoscaling is making the log file as "*/tmp/autoscaling.log*", you should check the size of it and in case you can remove it. You may avoid any log creation using "--nolog" (not recommended)

Autoscaling will execute as "cluster aware" if you provide a valid writable cluster filesystem (ACFS) and will consider the **cluster nodes load average**.

- Once autoscaling is firing a scale up/down request, DB System needs about 3 minutes to complete the operation. You need to consider this when setting the parameters like "--ocpu" & "--interval"
- Note: When using a low "--ocpu" scaling factor you may need multiple cycles to achieve the scaling target value to satisfy the incoming load. You may need to consider an higher scale factor instead.
- The '--dryrun' option will execute Autoscaling without performing any scale up/down of OCPUs
- DB System requires a minimum scaling factor ("--ocpu") multiple of Number of Compute Nodes
  - Exadata Quarter Rack = 2
  - Exadata Half Rack = 4
  - Exadata full Rack = 8
- OCPU are for all your nodes. If you have a 4 node ExaCS with OCPU=20, every single node will show (ie. using /proc/cpuinfo) 10 cpus due to Intel hyperthreading -->  $10/2 \times 4 = 20$
- While we immediately scale up the OCPUs in the event of high CPU utilization, we don't immediately drop the OCPUs should the load drop and CPU utilization drops below the target range. If the load is fluctuating we don't want to drop the OCPU on every dip in load as that will lead to more situations where the CPU is below target and service levels could be affected. Autoscaling will prevent the system from scaling down immediately. Rather, it will wait 120 minutes before lowering the OCPUs. This delay only takes effect after the CPU utilization has stabilized into the target range.



- On **ExaCC**, using "-vm-cluster-id", additional options may be required such "--ociregion" and "--proxy"
- Using "--ociregion <DB System region>" option, valid region are:

```
"ap-chuncheon-1"
"ap-melbourne-1"
"ap-mumbai-1"
"ap-hyderabad-1"
"ap-osaka-1"
"ap-seoul-1"
"ap-sydney-1"
"ap-tokyo-1"
"ca-montreal-1"
"ca-toronto-1"
"me-dubai-1"
"me-jeddah-1"
"eu-amsterdam-1"
"eu-frankfurt-1"
"eu-zurich-1"
"sa-saopaulo-1"
"uk-london-1"
"uk-cardiff-1"
"us-ashburn-1"
"us-phoenix-1"
"us-sanjose-1"
"us-seattle-1"
```

### Autoscaling Know Problem

1. Using "--acfs <writable cluster filesystem path>", Autoscaling is creating a file ".as-node\_<hostname>.json" describing the host load. If for any reason a node is crashing or if Autoscaling is killed (SIGKILL), such file is not removed and the Autoscaling running on other nodes is considering such node as a live node with load. The number of the nodes for which Autoscaling is measuring the load is showed on the log file, example

```
2020-12-11 10:56:57: Local host load .....: 6.1
2020-12-11 10:56:57: Cluster (4 node) Load : 16.7
2020-12-11 10:56:57: Next measure in 30 secs...
```

### Workaround

You can manually remove the ".as-node\_<hostname>.json" for the missing cluster node or for the node where Autoscaling has been killed with SIGKILL.

2. If you start autoscaling using 'root' pay attention to "/tmp/autoscaling.log" permissions, if later you need to start autoscaling with different user.

### Internals

- Using "--acfs <writable cluster filesystem path>" (cluster aware) option, Autoscaling will make following files under ACFS filesystem:

- .as-scalingtime.json
- .as-node\_<hostname>.json

File '.as-scalingtime.json' is describing latest executed scale-Up date time and by which host

File '.as-node\_<hostname>.json' is describing the host load

- Using unpublished "--debug" option, Autoscaling will make the "/tmp/autoscaling.log" more verbose

- Using "--ociregion <DB System region>" option, you are going to override the DB System region information, valid region are:

```
"ap-chuncheon-1"
"ap-melbourne-1"
"ap-mumbai-1"
"ap-hyderabad-1"
"ap-osaka-1"
"ap-seoul-1"
"ap-sydney-1"
"ap-tokyo-1"
"ca-montreal-1"
"ca-toronto-1"
"me-dubai-1"
"me-jeddah-1"
"eu-amsterdam-1"
"eu-frankfurt-1"
```

```
"eu-zurich-1"
"sa-saopaulo-1"
"uk-london-1"
"uk-cardiff-1"
"us-ashburn-1"
"us-phoenix-1"
"us-sanjose-1"
"us-seattle-1"
```

- Using "**--shape <DB System shape>**" option, you are going to override the DB System shape information, valid shape are:

```
"ExadataCC.FullX8M.400",
"ExadataCC.HalfX8M.200",
"ExadataCC.QuarterX8M.100",
"ExadataCC.BaseX8M.48",
"ExadataCC.Full3.400",
"ExadataCC.Half3.200",
"ExadataCC.Quarter3.100",
"ExadataCC.Base3.48",
"ExadataCC.Full12.368",
"ExadataCC.Half2.184",
"ExadataCC.Quarter2.92",
"ExadataCC.Base2.44",
"Exadata.X8M",
"Exadata.Full3.400",
"Exadata.Half3.200",
"Exadata.Quarter3.100",
"Exadata.Full12.368",
"Exadata.Half2.184",
"Exadata.Quarter2.92",
"Exadata.Full11.336",
"Exadata.Half1.168",
"Exadata.Quarter1.84",
"Exadata.Base.48"
```

[This section is not visible to customers.]

REFERENCES

<https://docs.cloud.oracle.com/en-us/iaas/Content/Functions/Tasks/functionssetupapikey.htm>  
<https://docs.cloud.oracle.com/en-us/iaas/Content/Database/References/exahardwareshapeconfig.htm>  
Didn't find what you are looking for?

Document Attributes

Author: RCITTON.IT;	Status: REVIEW_READY(INTERNAL) ***Oracle Confidential - Internal Use Only***
Owner: RCITTON.IT	Publisher: KFEKETE.HU
Alias:	Content Type: TEXT/X-HTML
Distribution: INTERNAL	Visibility: <b>INTERNAL</b>
Created By: RCITTON.IT	Created: Oct 14, 2020
Modified By: RCITTON.IT	Modified: Feb 12, 2021
Reviewed By: KFEKETE.HU	Reviewed: Feb 16, 2021
Source: AWIZ	Exception: No
Priority: 3	