

```

    <dbname>          : *Database name
    <dbtemplate>      : Default: ADMIN, otherwise ADMIN or DW, not required for
PDB
    <dbtype>          : Default: normal DB, otherwise CDB or PDB
    <hostnames>       : Default: list of nodes from db home
    <recodg>          : *The RECO diskgroup, not required for PDB
Where:
    <databaseid>      : The es xml id of the database
    <clusternumber>    : The cluster number in the es xml, starting at 1
    <clustername>      : The name of the cluster
    <clusterid>        : The es xml id of the cluster
    <databasename>     : The name of the database in the cluster

```

4.3 OEDACLI Command Reference

These OEDACLI commands are arranged by the object they act upon.

- **ACFSVOLUME**
You can add, alter, delete, or list Oracle ACFS volumes.
- **ACTION**
You can save, merge, and deploy OEDACLI actions.
- **ALERT**
You can create, alter, delete, or list Oracle alerts.
- **CELL**
You can clone, delete, or list Oracle Exadata storage cells.
- **CLUSTER**
You can alter or list the Oracle RAC Clusters on the Oracle Exadata Database Machine.
- **COMPUTE**
Use these commands to add or delete a physical compute node as a cluster member.
- **CURRENTACTION**
You can clear or list the current action in OEDACLI.
- **DATABASE**
You can add, alter, delete, or list the Oracle databases.
- **DATABASEHOME**
You can add, alter, delete, or list the Oracle Database homes.
- **DISKGROUP**
You can add, alter, delete, or list the Oracle ASM disk groups.
- **DOM0**
You can list Oracle VM management domains (dom0s).
- **DOMAIN**
You can get live information from a domain.
- **ES**
You can discover an engineered system or alter a limited number of attributes of an engineered system.
- **EXASCALE**
- **EXASCALECLUSTER**
- **EXITONERROR**
You can set the exit status for OEDACLI.

- **FILE**
You can manage the Engineered System XML file (`es.xml`).
- **GUEST**
You can clone or delete an Oracle VM guest.
- **ILOM**
You can alter or list the ILOMS in the loaded `es.xml` file.
- **LOGIN**
You can check remote login to a machine or cluster.
- **MACHINE**
You can alter or list the machines in the `es.xml` file.
- **NETWORK**
You can add, alter, delete, or list the networks defined in the XML configuration file.
- **PASSWORD**
You can specify the default passwords.
- **PROPERTY**
You can set OEDA properties.
- **RACK**
You can manage an Engineered System rack.
- **REQUIREDFILES**
You can display information about required files.
- **REMOTE**
You can run a command on a remote host.
- **SCAN**
You can add, alter, delete, or list the SCANS for a cluster.
- **SECURITY**
You can manage security settings.
- **SOFTWAREVERSIONS**
You can check supported software versions.
- **SSHKEYS**
You can generate and deploy SSH keys for `root` user authentication on Oracle Exadata Rack.
- **STEPS**
You can display the steps for a virtual `CREATE CLUSTER` deployment.
- **STORAGEPOOL**
- **STORAGETEMPLATES**
- **SWITCH**
You can display and manage the switches.
- **SU**
Controls use of the `su` command.
- **ULOC**
You can display the location of rack items.
- **VAULT**
- **VIP**
You can add, alter, delete, or list the Oracle Clusterware VIPs.

- **VOLUME**
You can display and manage storage volumes.
- **XMLACTION**
You can manage actions relating to the Engineered System XML file.

4.3.1 ACFSVOLUME

You can add, alter, delete, or list Oracle ACFS volumes.

- **ADD ACFSVOLUME**
- **ALTER ACFSVOLUME**
- **DELETE ACFSVOLUME**
- **LIST ACFSVOLUMES**

4.3.1.1 ADD ACFSVOLUME

This command adds Oracle Advanced Cluster File System (Oracle ACFS) volumes.

Syntax

```
ADD ACFSVOLUME
  VOLUMENAME=volume_name
  VOLUMESIZE=volume_size
  MOUNTPATH=mount_path
WHERE
  { DISKGROUPID=diskgroup_id |
    CLUSTERNUMBER=cluster_number DISKGROUPNAME=diskgroup_name |
    CLUSTERNAME=cluster_name DISKGROUPNAME=diskgroup_name |
    CLUSTERID=cluster_id DISKGROUPNAME=diskgroup_name }
```

Arguments

Specify the following attributes for the new Oracle ACFS volume:

- **VOLUMENAME** : Specifies a unique name for the new volume
- **VOLUMESIZE** : A number that specifies the size of the new volume in GB
- **MOUNTPATH** : Specifies the file system mount path for the new volume

The following arguments are available in the **WHERE** clause:

- **DISKGROUPID** : Specifies the *es.xml* ID for the container disk group
- **CLUSTERNUMBER** : Specifies the cluster number in the *es.xml*, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the *es.xml* ID of the cluster
- **DISKGROUPNAME** : Specifies the name of the container disk group

4.3.1.2 ALTER ACFSVOLUME

This command alters an existing Oracle Advanced Cluster File System (Oracle ACFS) volume.

Syntax

```

ALTER ACFSVOLUME
  { VOLUMENAME=volume_name |
    VOLUMESIZE=volume_size |
    MOUNTPATH=mountpath } ...
WHERE
  { ID=acfs_volume_id |
    DISKGROUPID=diskgroup_id VOLUMENAME=volume_name |
    CLUSTERNUMBER=cluster_number DISKGROUPNAME=diskgroup_name
    VOLUMENAME=volume_name |
    CLUSTERNAME=cluster_name DISKGROUPNAME diskgroup_name
    VOLUMENAME=volume_name |
    CLUSTERID=cluster_id DISKGROUPNAME=diskgroup_name
    VOLUMENAME=volume_name }

```

Arguments

You can modify the following attributes of the Oracle ACFS volume:

- **VOLUMENAME** : Specifies a new, unique volume name
- **VOLUMESIZE** : A number that specifies the new size of the volume in GB
- **MOUNTPATH** : Specifies the new file system mount path for the volume

The following arguments are available in the **WHERE** clause:

- **ID** : Specifies the `es.xml` ID for an existing Oracle ACFS volume
- **DISKGROUPID** : Specifies the `es.xml` ID for the container disk group
- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster
- **DISKGROUPNAME** : Specifies the name of the container disk group
- **VOLUMENAME** : Specifies the name of the existing Oracle ACFS volume

4.3.1.3 DELETE ACFSVOLUME

This command deletes an existing Oracle Advanced Cluster File System (Oracle ACFS) volume.

Syntax

```

DELETE ACFSVOLUME
WHERE
  { ID=acfs_volume_id |
    DISKGROUPID=diskgroup_id VOLUMENAME=volume_name |
    CLUSTERNUMBER=cluster_number DISKGROUPNAME=diskgroup_name
    VOLUMENAME=volume_name |
    CLUSTERNAME=cluster_name DISKGROUPNAME=diskgroup_name
    VOLUMENAME=volume_name |

```

```
CLUSTERID=cluster_id DISKGROUPNAME=diskgroup_name
VOLUMENAME=volume_name }
```

Arguments

The following arguments are available in the WHERE clause:

- ID : Specifies the `es.xml` ID for the Oracle ACFS volume
- DISKGROUPID : Specifies the `es.xml` ID for the container disk group
- CLUSTERNUMBER : Specifies the cluster number in the `es.xml`, starting at 1
- CLUSTERNAME : Specifies the name of the cluster
- CLUSTERID : Specifies the `es.xml` ID of the cluster
- DISKGROUPNAME : Specifies the name of the container disk group
- VOLUMENAME : Specifies the name of the existing Oracle ACFS volume

4.3.1.4 LIST ACFSVOLUMES

This command lists the Oracle Advanced Cluster File System (Oracle ACFS) volumes.

Syntax

```
LIST ACFSVOLUMES
[ WHERE
    { ID=acfs_volume_id |
      DISKGROUPID=diskgroup_id |
      CLUSTERNUMBER=cluster_number [ DISKGROUPNAME=diskgroup_name ] |
      CLUSTERNAME=cluster_name [ DISKGROUPNAME=diskgroup_name ] |
      CLUSTERID=cluster_id [ DISKGROUPNAME=diskgroup_name ] }
```

Arguments

The following arguments are available in the WHERE clause:

- ID : Specifies the `es.xml` ID for the Oracle ACFS volume
- DISKGROUPID : Specifies the `es.xml` ID for the container disk group
- CLUSTERNUMBER : Specifies the cluster number in the `es.xml`, starting at 1
- CLUSTERNAME : Specifies the name of the cluster
- CLUSTERID : Specifies the `es.xml` ID of the cluster
- DISKGROUPNAME : Specifies the name of the container disk group

4.3.2 ACTION

You can save, merge, and deploy OEDACLI actions.

- [SAVE ACTION](#)
- [MERGE ACTIONS](#)
- [DEPLOY ACTIONS](#)

4.3.2.1 SAVE ACTION

Use this command to save the current action to in-memory copy of the Engineered System XML configuration.

Syntax

```
SAVE ACTION [FORCE]
```

Arguments

Specify **FORCE** to auto-merged the action into the Engineered System model.



Note:

The Engineered System configuration should be saved as a file using a subsequent **SAVE FILE** command.

4.3.2.2 MERGE ACTIONS

Use this command to merge the current actions into the Engineered System model.

Syntax

```
MERGE ACTIONS [FORCE]
```

Arguments

Specify **FORCE** if you do not want this action to be processed by a **DEPLOY ACTIONS** command.



Note:

The Engineered System configuration should be saved as a file using a subsequent **SAVE FILE** command.

4.3.2.3 DEPLOY ACTIONS

Use this command to run all actions from the Engineered System XML file.

Syntax

```
DEPLOY ACTIONS
```

Example 4-3 Deploying the Actions to Add a Database

```
oedaccli> DEPLOY ACTIONS
Deploying Action ID : 4 add database dbname='testdb' DATADG='DATA1'
RECOG='RECOC1' where DBHOMEID='c0_databaseHome'
Deploying ADD DATABASE
```

```
Running datapatch on database 'testdb'
Done...
Done
```

4.3.3 ALERT

You can create, alter, delete, or list Oracle alerts.

- [ALTER ALERT](#)
- [DELETE ALERT](#)
- [LIST ALERTS](#)

4.3.3.1 ALTER ALERT

This command creates or alters an Oracle alert.

Syntax

```
ALTER ALERT
  { RECIPIENTS='recipients'
    FROMNAME='from_name'
    FROMEMAIL=from_email
    SMTPHOST=smtp_host
    [ SSL=uses_ssl ]
    [ PORT=smtp_port ]
  | SNMPHOST=snmp_host
    [ PORT=snmp_port ]
    [ COMMUNITY=snmp_community ]
  | ASRHOST=asr_host
    CONTACT=asr_contact
    EMAIL=asr_email
    MOSACCOUNT=asr_mos_account
    [ PROXYHOST=asr_proxy_host ]
    [ PROXYPORT=asr_proxy_port ]
    [ PROXYUSER=asr_proxy_user ]
  | [ EMAIL=ocm_email ]
    [ HUBHOST=ocm_hub_host ]
    [ HUBUSER=ocm_hub_user ]
    [ PROXYHOST=ocm_proxy_host ]
    [ PROXYPORT=ocm_proxy_port ]
    [ PROXYUSER=ocm_proxy_user ]
  | EMBASE=em_base
    [ OMSHOST=oms_host ]
    PORT=oms_port }
WHERE
  { ID=alert_id |
    TYPE=alert_type |
    HOSTNAME=host_name TYPE=alert_type |
    CLUSTERNUMBER=cluster_number TYPE=alert_type |
    CLUSTERNAME=cluster_name TYPE=alert_type |
    CLUSTERID=cluster_id TYPE=alert_type }
```

Arguments

You can specify various attributes according to the alert type.

- For SMTP alerts:
 - `RECIPIENTS` : Specifies a list of email addresses. The list entries are separated by a space or a comma and the list is surrounded by single quotation marks
 - `FROMNAME` : Specifies the name of the email sending user, surrounded by single quotation marks
 - `FROMEMAIL` : Specifies the email address of the email sending user
 - `SMTPHOST` : Specifies the hostname of the SMTP email server
 - `SSL` : Optionally, set to `true` to use SSL. Default value is `false`.
 - `PORT` : Optionally, specifies the SMTP email server port. Default value is 25.
- For SNMP alerts:
 - `SNMPHOST` : Specifies the hostname of the SNMP server
 - `PORT` : Optionally, specifies the SNMP server port. Default value is 162.
 - `COMMUNITY` : Optionally, specifies the SNMP community value. Default value is `public`.
- For Oracle Auto Service Request (ASR) alerts:
 - `ASRHOST` : Specifies the hostname of the ASR server
 - `CONTACT` : Specifies the ASR contact name
 - `EMAIL` : Specifies the ASR email address
 - `MOSACCOUNT` : Specifies the ASR My Oracle Support (MOS) email address
 - `PROXYHOST` : Optionally, specifies the proxy hostname for connectivity to the ASR server
 - `PROXYPORT` : Optionally, specifies the proxy port for connectivity to the ASR server
 - `PROXYUSER` : Optionally, specifies the proxy username for connectivity to the ASR server
- For Oracle Configuration Manager (OCM) alerts:
 - `EMAIL` : Specifies the OCM email address
 - `HUBHOST` : Specifies the hostname of the OCM hub
 - `HUBUSER` : Specifies the OCM hub user name
 - `PROXYHOST` : Optionally, specifies the proxy hostname for connectivity to the OCM hub
 - `PROXYPORT` : Optionally, specifies the proxy port for connectivity to the OCM hub
 - `PROXYUSER` : Optionally, specifies the proxy username for connectivity to the OCM hub
- For Oracle Enterprise Manager (EM) alerts:
 - `EMBASE` : Optionally, specifies the EM base directory. Default value is `/u01/app/oracle/product/EMbase`.
 - `OMSHOST` : Specifies the hostname of the Oracle Management Server (OMS)
 - `PORT` : Optionally, specifies the OMS server port. Default value is 4900.

The following arguments are available in the `WHERE` clause:

- `ID` : Specifies the `es.xml` ID for an existing alert
- `TYPE` : Specifies the alert type: SNMP, SMTP, ASR, OCM, or EM
- `HOSTNAME` : Specifies the target server hostname
- `CLUSTERNUMBER` : Specifies the cluster number in the `es.xml`, starting at 1
- `CLUSTERNAME` : Specifies the name of the cluster
- `CLUSTERID` : Specifies the `es.xml` ID of the cluster

4.3.3.2 DELETE ALERT

This command deletes a specific alert.

Syntax

```
DELETE ALERT
WHERE
  { ID=alert_id |
    HOSTNAME=host_name [ TYPE=alert_type ] |
    CLUSTERNUMBER=cluster_number [ TYPE=alert_type ] |
    CLUSTERNAME=cluster_name [ TYPE=alert_type ] |
    CLUSTERID=cluster_id [ TYPE=alert_type ] }
```

Arguments

The following arguments are available in the `WHERE` clause:

- `ID` : Specifies the `es.xml` ID for an existing alert
- `HOSTNAME` : Specifies the target server hostname
- `TYPE` : Specifies the alert type: SNMP, SMTP, ASR, OCM, or EM
- `CLUSTERNUMBER` : Specifies the cluster number in the `es.xml`, starting at 1
- `CLUSTERNAME` : Specifies the name of the cluster
- `CLUSTERID` : Specifies the `es.xml` ID of the cluster

4.3.3.3 LIST ALERTS

This command lists alerts.

Syntax

```
LIST ALERTS
[ WHERE
  { ID=alert_id |
    TYPE=alert_type |
    HOSTNAME=host_name [ TYPE=alert_type ] } ]
```

Arguments

The following arguments are available in the `WHERE` clause:

- ID : Specifies the `es.xml` ID for an existing alert
- TYPE : Specifies the alert type: `SNMP`, `SMTP`, `ASR`, `OCM`, or `EM`
- HOSTNAME : Specifies the target server hostname

4.3.4 CELL

You can clone, delete, or list Oracle Exadata storage cells.

- [CLONE CELL](#)
- [CLONE NEWCELL](#)
- [DELETE CELL](#)
- [DELETE NEWCELL](#)
- [LIST CELLS](#)

4.3.4.1 CLONE CELL

This command clones an existing storage cell.



Note:

This command is deprecated and replaced by `CLONE NEWCELL`.

Syntax

```
CLONE CELL
  SRCNAME=source_cell
  [ TGTNAME=new_cell_name ]
  [ WAIT=wait ]
  [ POWER=power ]
  [ WHERE STEPNAME=step ]

SET ADMINNET NAME=admin_name IP=admin_ip [ NETMASK=netmask ]
[ GATEWAY=gatewayip ]
SET PRIVNET NAME1=priv_name_1 IP1=priv_ip1 NAME2=priv_name_2 IP2=priv_ip2
SET ILOMNET NAME=ilom_name IP=ilom_ip [ NETMASK=netmask ]
[ GATEWAY=gatewayip ]
SET RACK NUM=rack_number ULOC=ulocation
```

Arguments

- SRCNAME: Specifies the name of an existing cell from which the new cell is derived.
- TGTNAME: Specifies the name of the new cell being added.
- WAIT: Default is `true`, in which case the command waits for completion of the ASM rebalance operation prior to completion. Optionally, set to `false` to return control without waiting for the ASM rebalance operation.
- POWER : Optionally, specifies the ASM rebalance power limit. The range of valid values is 1 - 1024.

- **STEPNAME:** Specifies the deployment step to perform, which is one of the following:

- CONFIG_CELL
- CREATE_GRIDDISKS
- ADD_DISKS_TO_ASM
- REBALANCE

You can set the following options for a cloned cell:

- **ADMINNET :** Specifies the DNS name and IP address to use for the administration network. The DNS name may optionally include the domain name. You can also optionally specify the netmask and gateway for the administration network.
- **PRIVNET :** Specifies the names and IP addresses to use for the first and second private networks. The names may optionally include the domain name.
- **ILOMNET :** Specifies the DNS name and IP address to use for the ILOM network. The DNS name may optionally include the domain name. You can also optionally specify the netmask and gateway for the ILOM network.
- **RACK :** Specifies the rack number in a multi-rack configuration, where 1 is the first rack, and the physical location (slot number) of the cell in the rack.

The following example shows the command sequence for adding cell `dbm01celadm14` by cloning `dbm01celadm09`.

```
# Clone
CLONE CELL SRCNAME=dbm01celadm09 TGTNAME=dbm01celadm14 where
STEPNAME=CONFIG_CELL
SET ADMINNET NAME=dbm01celadm14 IP=203.0.161.160
SET PRIVNET NAME1=dbm01celadm14-priv1 IP1=192.168.160.19 NAME2=dbm01celadm14-
priv2 IP2=192.168.160.20
SET ILOMNET NAME=dbm01celadm14-ilom IP=203.0.161.182
SET RACK NUM=1, ULOC=12

SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS

# Create Griddisks
CLONE CELL SRCNAME=dbm01celadm09 TGTNAME=dbm01celadm14 where
STEPNAME=CREATE_GRIDDISKS

SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS

# Add Griddisks to ASM Diskgroups
CLONE CELL SRCNAME=dbm01celadm09 TGTNAME=dbm01celadm14 where
STEPNAME=ADD_DISKS_TO_ASM

SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS

# Rebalance ASM Diskgroups
CLONE CELL SRCNAME=dbm01celadm09 TGTNAME=dbm01celadm14 where
```

```
STEPNAME=REBALANCE
```

```
SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
```

4.3.4.2 CLONE NEWCELL

This command clones an existing storage cell.

This command replaces the deprecated `CLONE CELL` command. Instead of using the `STEPNAME` argument, `CLONE NEWCELL` works in conjunction with the `ALTER CLUSTER ADDCELLS` command to add the new storage into an existing cluster.

Syntax

```
CLONE NEWCELL
  SRCNAME=source_cell
  TGTNAME=new_cell_name
  [ TYPE=cell_type ]

SET ADMINNET NAME=admin_name IP=admin_ip [ NETMASK=netmask ]
[ GATEWAY=gatewayip ]
SET PRIVNET NAME1=priv_name_1 IP1=priv_ip1 NAME2=priv_name_2 IP2=priv_ip2
SET ILOMNET NAME=ilom_name IP=ilom_ip [ NETMASK=netmask ]
[ GATEWAY=gatewayip ]
SET RACK NUM=rack_number ULOC=ulocation
```

Arguments

- **SRCNAME:** Specifies the name of an existing cell from which the new cell is derived.
- **TGTNAME:** Specifies the name of the new cell being added.
- **TYPE:** Optionally specifies the cell type. Use this option only for situations where the clone source is a high capacity (HC) storage server, and the target is an extended (XT) storage server. The following values are allowed:
 - **X8XT:** Specifies that the target is an XT storage server with InfiniBand Network Fabric.
 - **X8MXT:** Specifies that the target is an XT storage server with RoCE Network Fabric.

You can set the following options for a cloned cell:

- **ADMINNET :** Specifies the DNS name and IP address to use for the administration network. The DNS name may optionally include the domain name. You can also optionally specify the netmask and gateway for the administration network.
- **PRIVNET :** Specifies the names and IP addresses to use for the first and second private networks. The names may optionally include the domain name.
- **ILOMNET :** Specifies the DNS name and IP address to use for the ILOM network. The DNS name may optionally include the domain name. You can also optionally specify the netmask and gateway for the ILOM network.
- **RACK :** Specifies the rack number in a multi-rack configuration, where 1 is the first rack, and the physical location (slot number) of the cell in the rack.

The following example shows the command sequence for adding cell `dbm02celadm06` by cloning `dbm02celadm01`.

```
# Clone
CLONE NEWCELL SRCNAME=dbm02celadm01 TGTNAME=dbm02celadm06
SET ADMINNET NAME=dbm02celadm06 IP=203.0.96.17
SET PRIVNET NAME1=dbm02cell06vm1str-priv1 IP1=192.168.3.47
NAME2=dbm02cell06vm1str-priv2 IP2=192.168.3.48
SET ILOMNET NAME=dbm02celadm06-ilom IP=203.0.10.103
SET RACK NUM=1 ULOC=12

SAVE ACTION

# Add Cell to Cluster
ALTER CLUSTER ADDCELLS='dbm02celadm06' POWER=4 WHERE CLUSTERNUMBER=1

SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
```

4.3.4.3 DELETE CELL

This command deletes a cell from the OEDA XML configuration file (`es.xml`).

Syntax

```
DELETE CELL
WHERE
    SRCNAME=cell_hostname
```

Arguments

- `SRCNAME` : Specifies the hostname of the cell being deleted.

Usage Notes

- The command performs no deployment actions, and no deployed cells are affected.
- Use `SAVE ACTION FORCE` to save changes to the XML configuration file.
- To remove a deployed cell, use the `ALTER CLUSTER` command instead of `DELETE CELL`.

4.3.4.4 DELETE NEWCELL

This command deletes a cell from an OEDA configuration file (`es.xml`).

Syntax

```
DELETE NEWCELL
WHERE
    { SRCNAME=cell_host_name |
      SRCNAMES=cell_host_names }
```

Arguments

WHERE clause arguments:

- SRCNAME : Specifies the host name of the cell being deleted from `es.xml`.
- SRCNAMES : Specifies a comma-separated list of host names for the cells being deleted from `es.xml`.

Usage Notes

Use `SAVE ACTION FORCE` to save changes only to `es.xml`. No deployment actions are performed.

4.3.4.5 LIST CELLS

This command lists the storage cells.

Syntax

```
LIST CELLS
```

Usage Notes

This command is deprecated. Instead, use:

```
LIST MACHINES WHERE TYPE=CELL
```

4.3.5 CLUSTER

You can alter or list the Oracle RAC Clusters on the Oracle Exadata Database Machine.

- [ALTER CLUSTER](#)
- [CREATE CLUSTER](#)
Use this command to create virtual guests for a virtual machine (VM) cluster, including the steps to configure and deploy an Oracle RAC database.
- [DESTROY CLUSTER](#)
This command is used to cleanly remove a running virtual machine (VM) cluster in one operation.
- [DOWNGRADE CLUSTER](#)
Use this command downgrade a previously upgraded cluster.
- [LIST CLUSTER](#)
This command lists a specific Exadata cluster.
- [LIST CLUSTERS](#)
This command lists the existing Exadata clusters.
- [UPGRADE CLUSTER](#)
Use this command upgrade an Oracle Grid Infrastructure cluster.

4.3.5.1 ALTER CLUSTER

This command alters Oracle Grid Infrastructure (GI) cluster attributes.

Syntax

```
ALTER CLUSTER
  { CLUSTERNAME=cluster_name |
    GIVERSION=grid_version |
    GIHOMELOC=grid_home_location |
    INVLOC=inventory_location |
    BASEDIR=oracle_base_dir |
    PATCHLIST=patch_list |
    VCPU=vcpu |
    VMEM=vmem |
    ASMSCOPEDSECURITY=asm_scoped_security |
    HOMESIZE=home_size |
    INSTALLTYPE={ CLONE | ZIP }
  } ...
WHERE
  { CLUSTERNUMBER=cluster_number |
    CLUSTERNAME=cluster_name |
    ID=cluster_id }

ALTER CLUSTER
  { ADDCELLS='add_cells' | DROPCELLS='drop_cells' }
  [ DISKGROUPS='asm_diskgroups' ] [ POWER=power ] [ WAIT={ TRUE |
FALSE }]
WHERE
  { CLUSTERNUMBER=cluster_number |
    CLUSTERNAME=cluster_name |
    ID=cluster_id }
  [ STEPNAME=step_name ]
```

Arguments

You can modify the following attributes of the GI cluster:

- **CLUSTERNAME:** Specifies a new, unique name for the cluster
- **GIVERSION:** Specifies the clusterware version, including the release update (RU). For example: 12.1.0.2.170418
- **GIHOMELOC:** Specifies a new path for the Grid home location
- **INVLOC:** Specifies a new path for the software inventory
- **BASEDIR:** Specifies a new path for the Oracle base directory
- **PATCHLIST:** Specifies a comma-separated list of patches. The list must be enclosed in single quotes. By default, the patch list is derived from the RU included in the **GIVERSION**.
- **VCPU:** Specifies the number of virtual CPUs for each virtual machine (VM) in a VM cluster. This option only applies to a cluster in a VM deployment.
- **VMEM:** Specifies the amount of memory (in GB) for each virtual machine (VM) in a VM cluster. This option only applies to a cluster in a VM deployment.
- **ASMSCOPEDSECURITY:** Specify **true** to enable ASM scoped security. The default value is **false**.
- **HOMESIZE:** Specifies the cluster home size (in GB).

- **INSTALLTYPE:** Specifies the software installation method to use. The options are:
 - **CLONE:** Uses image files to clone a software installation. The image files include the base software release, and may also include additional patches and updates. By including updates in the image files, this method reduces deployment time and improves deployment consistency.
The **CLONE** software installation method is the only supported method for a cluster in a virtual machine (VM) deployment. This option is also available for clusters in a physical (bare metal) deployment.
 - **ZIP:** Performs a fresh software installation by using compressed (zip) files that contain the software release. This method does not include additional patches or updates.
The **ZIP** software installation method is the default method for clusters in a physical (bare metal) deployment. This option is not available for clusters in a virtual machine (VM) deployment.
- **ADDCELLS:** Specifies a comma-separated list of cells to add to the disk groups. The list must be enclosed in single quotes. If multiple cells are specified, they are all added in one operation, and the data is rebalanced once at the end of the process.
- **DROPCELLS:** Specifies a comma-separated list of cells to drop from the disk groups. The list must be enclosed in single quotes. If multiple cells are specified, they are all dropped in one operation, and the data is rebalanced once at the end of the process.
- **DISKGROUPS:** Specifies a comma-separated list of disk groups from which you want to add or remove cells specified in **ADDCELLS** or **DROPCELLS**. The list must be enclosed in single quotes. If not specified, then the specified cells are added to or removed from all of the disk groups.
- **POWER:** Specifies the ASM rebalance power limit. The range of valid values is 1 - 1024.
- **WAIT:** Specifies whether or not to wait for the completion of the ASM rebalance operation that occurs during the **ADD_DISKS_TO_ASM** step of an **ADDCELLS** or **DROPCELLS** operation. This option is only applicable when the **WHERE** clause contains **STEPNAME=ADD_DISKS_TO_ASM**.
Specify **false** to proceed without waiting for the ASM rebalance operation to complete. Specify **true** to wait for the ASM rebalance operation to complete. The default value is **true**.

The following arguments are available in the **WHERE** clause:

- **CLUSTERNUMBER:** Specifies the cluster number in the **es.xml**, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **ID:** Specifies the **es.xml** ID of the cluster
- **STEPNAME:** Specifies one of the following steps for an **ADDCELLS** or **DROPCELLS** operation:
 - **CONFIG_CELL**
 - **CREATE_GRIDDISKS**
 - **ADD_DISKS_TO_ASM**

For an **ADDCELLS** operation, the steps are performed in the following order:

1. **CONFIG_CELL:** Configures the storage server, including cell connectivity and cell disk creation.
2. **CREATE_GRIDDISKS:** Creates grid disks for use by the specified cluster.

3. **ADD_DISKS_TO_ASM**: Adds the newly created grid disks to the ASM disk groups associated with the specified cluster.

For a **DROPCELLS** operation, the steps are performed in reverse order:

1. **ADD_DISKS_TO_ASM**: Removes the grid disks from the ASM disk groups associated with the specified cluster.
2. **CREATE_GRIDDISKS**: Deletes grid disks associated with the specified cluster.
3. **CONFIG_CELL**: Deletes the cell configuration on the storage server.

Example 4-4 Deploying a Cluster Change

The following example shows an OEDACLI session that uses the **ALTER CLUSTER** command along with typical output at each step.

```
oedaccli> alter cluster clustername=cluster-c1 where id=Cluster-c6e838fe4-e317-
ee1a-dd89-53f0b49a689b_id
oedaccli> save action
oedaccli> merge actions
processMerge
processMergeActions
Merging Action : alter cluster clustername=cluster-c1 where id=Cluster-
c6e838fe4-e317-ee1a-dd89-53f0b49a689b_id
Merging ALTER CLUSTER
Action Validated and Merged OK
oedaccli> deploy actions
Deploying Action ID : 1 alter cluster clustername=cluster-c1 where
id=Cluster-c6e838fe4-e317-ee1a-dd89-53f0b49a689b_id
Deploying ALTER CLUSTER
Done...
Done
```

4.3.5.2 CREATE CLUSTER

Use this command to create virtual guests for a virtual machine (VM) cluster, including the steps to configure and deploy an Oracle RAC database.

Syntax

```
CREATE CLUSTER
[ STEPS='step_short_names' [ DELETEDB={ true | false } ] |
  SKIPSTEPS='step_short_names' [ DELETEDB={ true | false } ] |
  DELETEDB={ true | false } ]
[ WHERE
  { CLUSTERNAME=cluster_name |
    CLUSTERNUMBER=cluster_number |
    ID=cluster_id } ]
```

Arguments

- **STEPS** : Optionally, specifies the steps to run.

The list consists of space-separated step short names and the entire list is enclosed in single quotes.

Use the **LIST STEPS** command to view the available steps.

- **SKIPSTEPS** : Optionally, specifies steps to skip.

The list consists of space-separated step short names and the entire list is enclosed in single quotes.

Use the `LIST STEPS` command to view the available steps.

- **DELETEDB** : Optionally, specify `true` to delete the default Oracle RAC database after cluster creation. The default value is `false`.

The following arguments are available in the `WHERE` clause:

- **CLUSTERNUMBER**: Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME**: Specifies the name of the cluster
- **CLUSTERID**: Specifies the `es.xml` ID of the cluster

Usage Notes

- The command can only create a cluster in a VM deployment. You cannot use this command to create the cluster in a physical (bare-metal) deployment.
- The simplest form of this command does not require any arguments. However, if the XML configuration file (`es.xml`) contains more than one cluster, then the `WHERE` clause is mandatory.

Example 4-5 Using the STEPS Parameter with CREATE CLUSTER

Use the following command to only create the guest domains for the `Cluster-c1` cluster:

```
CREATE CLUSTER STEPS='CREATEVM' WHERE CLUSTERNAME=Cluster-c1
```

Example 4-6 Using the SKIPSTEPS and DELETEDB Parameters with CREATE CLUSTER

Use the following command to create the `Cluster-c1` cluster, skipping the specified `SKIPSTEPS` and deleting the default database.

```
CREATE CLUSTER SKIPSTEPS='VALIDATE CALIBRATE RESECURE' DELETEDB=true WHERE  
CLUSTERNAME=Cluster-c1
```

4.3.5.3 DESTROY CLUSTER

This command is used to cleanly remove a running virtual machine (VM) cluster in one operation.

Syntax

```
DESTROY CLUSTER  
[ WHERE  
  { CLUSTERNUMBER=cluster_number |  
    CLUSTERNAME=cluster_name |  
    CLUSTERID=cluster_id } ]
```

Arguments

The following arguments are available in the `WHERE` clause:

- **CLUSTERNUMBER:** Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **CLUSTERID:** Specifies the `es.xml` ID of the cluster

Usage Notes

- No **WHERE** clause is required if the XML file contains one cluster only.
- The **MERGE** step indicates the machines that will be destroyed and the affected cells.

Example 4-7 Using the DESTROY CLUSTER Command

The following example shows the simplest case, where there is only a single cluster in the XML file.

```
oedaccli> DESTROY CLUSTER
oedaccli> SAVE ACTION
oedaccli> MERGE ACTIONS
oedaccli> DEPLOY ACTIONS
```

4.3.5.4 DOWNGRADE CLUSTER

Use this command downgrade a previously upgraded cluster.

Syntax

```
DOWNGRADE CLUSTER
WHERE
  { CLUSTERNAME=cluster_name [ STEPNAME=step_name ] |
    CLUSTERNUMBER=cluster_number [ STEPNAME=step_name ] |
    ID=cluster_id [ STEPNAME=step_name ] }
```

Arguments

The following arguments are available in the **WHERE** clause:

- **CLUSTERNUMBER:** Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **ID:** Specifies the `es.xml` ID of the cluster
- **STEPNAME :** Optionally, specifies the individual downgrade step:
 1. `RUN_ROOTSCRIPT`
 2. `CONFIG_HOME`
 3. `ADD_HOME`

Usage Notes

- To use the command, you must use the `es.xml` that contains the upgraded cluster.
- The command can only downgrade a cluster in a VM deployment. You cannot use this command to downgrade the cluster in a physical (bare-metal) deployment.
- The command can only run if the `DELETE_OLDHOME` step has not been run on the upgraded cluster.

Example 4-8 Cluster Downgrade

The following example shows an OEDACLI session that uses the `DOWNGRADE CLUSTER` command along with typical output at each step.

```
oedaccli> downgrade cluster where clusternumber=1
oedaccli> save action
oedaccli> merge actions
    processMerge
    processMergeActions
    Merging Action : downgrade cluster where clusternumber=1
    Merging DOWNGRADE CLUSTER
    Action Validated and Merged OK
oedaccli> deploy actions
    Deploying Action ID : 4 downgrade cluster where clusternumber=1
    Deploying DOWNGRADE CLUSTER
    Downgrading Cluster
    Validating Target cluster version 12.2.0.1
    Validating Target Clusterware Home.. /u01/app/12.2.0.1/grid
    Checking Cluster status...
    Checking status of cluster...
    Checking clusterware file system /u01/app/12.2.0.1/grid
    Validating active cluster version.
    Active cluster version found on cluster ovmClus8 is 18.0.0.0
    Validating Inventory...
    Clusterware Home /u01/app/12.2.0.1/grid validation completed successfully
    Downgrading clusterware to 12.2.0.1
    Running rootcrs.sh on node dbm01adm01vm08.example.com
    Running rootcrs.sh on node dbm01adm02vm08.example.com
    Deregistering Clusterware Home /u01/app/18.13.0.0/grid from oracle inventory
    Registering Clusterware Home /u01/app/12.2.0.1/grid with oracle inventory
    Starting Clusterware stack using software home /u01/app/12.2.0.1/grid
    Removing /u01/app/18.13.0.0/grid from Oracle inventory...
    Updating inventory on dbm01adm01vm08.example.com
    Updating inventory on dbm01adm02vm08.example.com
    Unmounting file systems.....
    Unmounting file system /u01/app/18.13.0.0/grid on dbm01adm01vm08.example.com
    Unmounting file system /u01/app/18.13.0.0/grid on dbm01adm02vm08.example.com
    Updating /etc/fstab entries...
    Active cluster version after downgraded 12.2.0.1.0
    Clusterware successfully downgraded to 12.2.0.1.190115
```

4.3.5.5 LIST CLUSTER

This command lists a specific Exadata cluster.

Syntax

```
LIST CLUSTER
WHERE
    { NAME=cluster_name |
      ID=cluster_id }
```

Arguments

The following arguments are available in the `WHERE` clause:

- **NAME:** Specifies the name of the cluster
- **ID:** Specifies the `es.xml` ID of the cluster

4.3.5.6 LIST CLUSTERS

This command lists the existing Exadata clusters.

Syntax

```
LIST CLUSTERS
[ WHERE
  { CLUSTERNUMBER=cluster_number |
    CLUSTERNAME=cluster_name |
    ID=cluster_id } ]
```

Arguments

The following arguments are available in the `WHERE` clause:

- **CLUSTERNUMBER:** Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **ID:** Specifies the `es.xml` ID of the cluster

Usage Notes

The `WHERE` clause is optional.

Example 4-9 Using the LIST CLUSTERS Command

This example shows how to identify the Oracle RAC clusters on the host.

```
OEDACLI> LIST CLUSTERS
version : "CloneInstall"
clusterName : "DBMclul"
clusterOwner : "grid"
clusterVersion : "19.3.0.0.190416"
clusterHome : "/u01/app/19.0.0.0/grid"
inventoryLocation : "/u01/app/oraInventory"
asmScopedSecurity : "false"
```

4.3.5.7 UPGRADE CLUSTER

Use this command upgrade an Oracle Grid Infrastructure cluster.

Syntax

```
UPGRADE CLUSTER
  GIVERSION=Grid_relnum
  GIHOMELOC=Grid_home_dir
```

```
WHERE
  { CLUSTERNAME=cluster_name [ STEPNAME=step_name ] |
    CLUSTERNUMBER=cluster_number [ STEPNAME=step_name ] |
    ID=cluster_id [ STEPNAME=step_name ] }
```

Arguments

- **GIVERSION:** The version of the Oracle Grid Infrastructure software you are upgrading to, including the release update (RU). For example: 12.1.0.2.170418
- **GIHOMELOC:** The target location of the Oracle Grid Infrastructure home directory

The following arguments are available in the `WHERE` clause:

- **CLUSTERNUMBER:** Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **ID:** Specifies the `es.xml` ID of the cluster
- **STEPNAME :** Optionally, specifies the individual upgrade step:
 1. `ADD_HOME`
 2. `CONFIG_HOME`
 3. `RUN_ROOTSCRIPT`
 4. `DELETE_OLDHOME`

Usage Notes

- The command can only upgrade a cluster in a VM deployment. You cannot use this command to upgrade the cluster in a physical (bare-metal) deployment.
- If `STEPNAME` is not specified, then the `ADD_HOME` and `CONFIG_HOME` steps are performed.
- The `RUN_ROOTSCRIPT` step runs the `rootupgrade.sh` script sequentially in a rolling mode on all nodes. This step also restarts the database instances on each of the nodes.
- After you run the `DELETE_OLDHOME` step in conjunction with the `UPGRADE CLUSTER` command, you cannot later use the `DOWNGRADE CLUSTER` command to roll back the upgrade.

Example 4-10 Using the UPGRADE CLUSTER Command

This example shows the first three steps (`ADD_HOME`, `CONFIG_HOME`, and `RUN_ROOTSCRIPT`) in a step-by-step cluster upgrade. At the end of this example, the cluster is upgraded, but downgrade is still possible because the `DELETE_OLDHOME` step has not been run.

```
oedaccli> upgrade cluster giversion=18.13.0.0.210119 gihome loc=/u01/app/
18.13.0.0/grid where clusternumber=1 stepname=add_home
oedaccli> save action
oedaccli> merge actions
processMerge
processMergeActions
Merging Action : upgrade cluster giversion=18.13.0.0.210119
gihome loc=/u01/app/18.13.0.0/grid where clusternumber=1 stepname=add_home
Merging UPGRADE CLUSTER
Action Validated and Merged OK
oedaccli> deploy actions
Deploying Action ID : 1 upgrade cluster giversion=18.13.0.0.210119
gihome loc=/u01/app/18.13.0.0/grid where clusternumber=1 stepname=add_home
```

```

Deploying UPGRADE CLUSTER
Upgrading Cluster
Performing Upgrade Validations...
Validating target Clusterware version 18.13.0.0...
Validating source Clusterware version 12.2.0.1.190115...
Checking Cluster status...
Checking status of cluster...
Active cluster version on cluster ovmClus8 is 12.2.0.1.0
Validating Target Clusterware Home.. /u01/app/18.13.0.0/grid
Checking to see if /u01/app/18.13.0.0/grid is already mounted
SUCCESS: 18.13.0.0 is supported version for upgrade.
SUCCESS: Current Cluster Version 12.2.0.1.190115 is supported for Upgrade
using this Utility.
SUCCESS:
SUCCESS: Required Image Version 21.2.0.0.0 for Oracle18c found on
dbm01adm01vm08
SUCCESS: Required Image Version 21.2.0.0.0 for Oracle18c found on
dbm01adm02vm08
SUCCESS: Cluster verification completed successfully
SUCCESS: Find the valid upgrading version from 12.2.0.1 to 18.13.0.0
SUCCESS: Target Cluster Version is valid..
SUCCESS: Target Clusterware Home /u01/app/18.13.0.0/grid is not already
mounted.
SUCCESS: Found valid target clusterware version.
SUCCESS: Valid Bundle Patch / RU Version found.
Validating required files...
Extracting files..
Copying files...
Creating new disk image file..
Attaching disk image to Virtual Machine dbm01adm01vm08.example.com
Attaching disk image to Virtual Machine dbm01adm02vm08.example.com
Completed adding new Clusterware home /u01/app/18.13.0.0/grid on Cluster
ovmClus8
Done...
Done
oedacli> upgrade cluster giversion=18.13.0.0.210119 gihome loc=/u01/app/
18.13.0.0/grid where clusternumber=1 stepname=config_home
oedacli> save action
oedacli> merge actions
processMerge
processMergeActions
Merging Action : upgrade cluster giversion=18.13.0.0.210119
gihome loc=/u01/app/18.13.0.0/grid where clusternumber=1 stepname=config_home
Merging UPGRADE CLUSTER
Action Validated and Merged OK
oedacli> deploy actions
Deploying Action ID : 2 upgrade cluster giversion=18.13.0.0.210119
gihome loc=/u01/app/18.13.0.0/grid where clusternumber=1 stepname=config_home
Deploying UPGRADE CLUSTER
Upgrading Cluster
Configuring new clusterware home at /u01/app/18.13.0.0/grid
Running Cluster Verification Utility for upgrade readiness..
Pre-upgrade Validation completed successfully.
Setting up new clusterware home...
Relinking binaries with RDS /u01/app/18.13.0.0/grid
Upgrading software..

```

```
Completed configuring new Clusterware home /u01/app/18.13.0.0/grid
Done...
Done
oedacli> upgrade cluster giversion=18.13.0.0.210119 gihomeloc=/u01/app/
18.13.0.0/grid where clusternumber=1 stepname=RUN_ROOTSCRIPT
oedacli> save action
oedacli> merge actions
processMerge
processMergeActions
Merging Action : upgrade cluster giversion=18.13.0.0.210119
gihomeloc=/u01/app/18.13.0.0/grid where clusternumber=1
stepname=RUN_ROOTSCRIPT
Merging UPGRADE CLUSTER
Action Validated and Merged OK
oedacli> deploy actions
Deploying Action ID : 3 upgrade cluster giversion=18.13.0.0.210119
gihomeloc=/u01/app/18.13.0.0/grid where clusternumber=1
stepname=RUN_ROOTSCRIPT
Deploying UPGRADE CLUSTER
Upgrading Cluster
Running Cluster Verification Utility for upgrade readiness..
Pre-upgrade Validation completed successfully.
Running rootupgrade.sh on node dbm01adm01vm08.example.com
Checking file
root_dbm01adm01vm08.example.com_2021-05-07_16-35-47-279774287.log on node
dbm01adm01vm08.example.com
Checking status of cluster on dbm01adm01vm08.example.com
Running rootupgrade.sh on node dbm01adm02vm08.example.com
Checking file
root_dbm01adm02vm08.example.com_2021-05-07_16-55-09-777507261.log on node
dbm01adm02vm08.example.com
Checking status of cluster on dbm01adm02vm08.example.com
Running Configuration Assistants on dbm01adm01vm08.example.com
Active cluster version after upgrade 18.0.0.0
Clusterware successfully upgraded to 18.13.0.0.210119
Applying Security Fixes...
Done...
Done
```

4.3.6 COMPUTE

Use these commands to add or delete a physical compute node as a cluster member.

- [CLONE COMPUTE](#)
Use this command to clone an existing physical compute cluster node.
- [DELETE COMPUTE](#)
This command deletes an existing compute node, removes the instance, home and clusterware from the cluster.

4.3.6.1 CLONE COMPUTE

Use this command to clone an existing physical compute cluster node.

Syntax

```
CLONE COMPUTE
  { SRCNAME=source_compute_node TGTNAME=target_compute_node |
    TGTNAMES=target_compute_nodes }
[ WHERE
  STEPNAME=stepname ]

SET ADMINNET NAME=admin_name IP=admin_ip
SET PRIVNET NAME1=priv_name_1 IP1=priv_ip1 [ NAME2=priv_name_2 IP2=priv_ip2 ]
[ SET INTERCONNECT NAME1=priv_name_1 IP1=priv_ip1 [ NAME2=priv_name_2
IP2=priv_ip2 ] ]
[ SET CLIENTNET NAME=client_name IP=client_ip ]
[ SET VIPNET NAME=vip_name IP=vip_ip ]
[ SET BACKUP NAME=backup_name IP=backup_ip ]
SET ILOMNET NAME=ilom_name IP=ilom_ip
SET RACK NUM=rack_number ULOC=ulocation
```

Arguments

- **SRCNAME:** Specifies the name of an existing compute server from which the compute server is derived.
- **TGTNAME:** Specifies the name of the new compute server to add.
- **TGTNAMES:** Specifies a comma-separated list of new compute servers to add.
- **STEPNAME:** Optionally specifies the deployment step to perform, which is one of the following:
 - CREATE_USERS
 - CELL_CONNECTIVITY
 - ADD_NODE
 - EXTEND_DBHOME
 - ADD_INSTANCE

If **STEPNAME** is not specified, then all steps are performed.

You can set the following options for a cloned compute node:

- **ADMINNET :** Specifies the DNS name and IP address to use for the administration network. The DNS name may optionally include the domain name.
- **PRIVNET :** Specifies the names and IP addresses to use for the first and second private networks. The names may optionally include the domain name.
- **INTERCONNECT:** Only required if the compute nodes are configured with Exadata Secure RDMA Fabric Isolation or InfiniBand security (PKEYS). Specifies the names and IP addresses to use for the first and second private networks. The names may optionally include the domain name.
- **CLIENTNET :** Specifies the DNS name and IP address to use for the client network. The DNS name may optionally include the domain name. Not required on virtual machine (VM) hosts (Oracle Linux KVM or Oracle VM).

- **VIPNET** : Specifies the DNS name and IP address to use for the virtual IP (VIP) network. The DNS name may optionally include the domain name. Not required on virtual machine (VM) hosts (Oracle Linux KVM or Oracle VM).
- **BACKUP** : Specifies the DNS name and IP address to use for the backup network. The DNS name may optionally include the domain name. Not required on servers without a backup network.
- **ILOMNET**: Specifies the DNS name and IP address to use for the ILOM network. The DNS name may optionally include the domain name.
- **RACK** : Specifies the rack number in a multi-rack configuration, where 1 is the first rack, and the physical location (slot number) of the server in the rack.

Usage Notes

- For configurations with Exadata Secure RDMA Fabric Isolation or InfiniBand security (PKEYS), **INTERCONNECT** defines the private network that connects the database nodes and **PRIVNET** defines the private storage network.
- Use **TGTNAMES** to clone multiple compute servers simultaneously in the **DEPLOY ACTIONS**. Use **TGTNAMES** without additional **SET** statements. Use **SAVE ACTION FORCE** prior to using this argument. See the example below.

Example 4-11 Cloning a Compute Node

This example clones the **dbadm13** compute server to a new server, **dbadm14**.

```
CLONE COMPUTE srcname=dbadm13 tgtname=dbadm14
SET ADMINNET name=dbadm04 ip=10.128.161.192
SET PRIVNET name1=dbadm04-priv1 ip1=192.168.161.195 name2=dbadm04-priv2
ip2=192.168.161.196
SET CLIENTNET name=dbadm04-client ip=198.51.100.112
SET VIPNET name=dbadm04-vip ip=10.128.177.116
SET ILOMNET name=dbadm04-ilom ip=10.128.161.203
SET RACK num=1 uloc=17

SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
```

Example 4-12 Cloning Multiple Compute Nodes Using TGTNAMES

This example clones **dbadm23** and creates the new compute servers **dbadm24** and **dbadm25**. By using the **TGTNAMES** argument, both new nodes are deployed at the same time in the final **DEPLOY ACTIONS**.

```
CLONE COMPUTE SRCNAME='dbadm23' TGTNAME='dbadm24'
SET ADMINNET name=dbadm24 ip=10.128.161.192
SET PRIVNET name1=dbadm24-priv1 ip1=192.168.161.195 name2=dbadm24-priv2
ip2=192.168.161.196
...
SAVE ACTION FORCE

CLONE COMPUTE SRCNAME='dbadm23' TGTNAME='dbadm25'
SET ADMINNET name=dbadm25 ip=10.128.161.193
...
SAVE ACTION FORCE
```

```
CLONE COMPUTE TGTNAMES='dbadm24,dbadm25'
```

```
SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
```

Example 4-13 Cloning a Compute Node Using Steps

This example clones the dbadm33 compute server to a new server, dbadm34, running individual steps. The example shows only the first three steps and does not show all the steps involved in cloning the compute node.

```
CLONE COMPUTE srcname=dbadm33 tgtname=dbadm34 WHERE STEPNAME=CREATE_USERS
SET ADMINNET name=dbadm04 ip=10.128.161.192
SET PRIVNET name1=dbadm04-priv1 ip1=192.168.161.195 name2=dbadm04-priv2
ip2=192.168.161.196
SET CLIENTNET name=dbadm04-client ip=198.51.100.112
SET VIPNET name=dbadm04-vip ip=10.128.177.116
SET ILOMNET name=dbadm04-ilom ip=10.128.161.203
SET RACK num=1 uloc=17
```

```
SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
```

```
CLONE COMPUTE compute srcname=dbadm33 tgtname=dbadm34 WHERE
STEPNAME=CELL_CONNECTIVITY
```

```
SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
```

```
CLONE COMPUTE srcname=dbadm33 tgtname=dbadm34 WHERE STEPNAME=ADD_NODE
```

```
SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
```

```
...
```

4.3.6.2 DELETE COMPUTE

This command deletes an existing compute node, removes the instance, home and clusterware from the cluster.

Syntax

```
DELETE COMPUTE
WHERE
  { SRCNAME=host_name [ STEPNAME=step_name ] |
    SRCNAMES=host_names [ STEPNAME=step_name ] }
```

Arguments

SRCNAME specifies the host name of the guest that you want to remove.

SRCNAMES specifies a comma-separated list host names of the guests that you want to remove.

STEPNAME is used to clean up steps from a failed `CREATE COMPUTE` command. Specify the name of the individual creation step to be reverted. The values for *step_name* are:

- `ADD_INSTANCE`
- `EXTEND_DBHOME`
- `ADD_NODE`
- `CELL_CONNECTIVITY`
- `CREATE_USERS`

Example 4-14 Deleting a Compute Node - Simple Case

This example removes the `dbadm04` compute node from the cluster.

```
LOAD FILE NAME=4-4compute.xml
DELETE COMPUTE where srcname=dbadm04
SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
SAVE FILE name=3-4compute.xml
```

Example 4-15 Deleting a Compute Node with Individual Steps

This example removes the `dbadm04` compute node from the cluster, running each step individually. This example does not show all the steps involved in removing the compute node.

```
LOAD FILE NAME=4-4compute.xml
DELETE COMPUTE where srcname=dbadm04 stepname=ADD_INSTANCE
SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS

DELETE COMPUTE where srcname=dbadm04 stepname=EXTEND_DBHOME
SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS

DELETE COMPUTE where srcname=dbadm04 stepname=ADD_NODE
SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
...
```

4.3.7 CURRENTACTION

You can clear or list the current action in OEDACLI.

- `CLEAR CURRENTACTION`

- [LIST CURRENTACTION](#)

4.3.7.1 CLEAR CURRENTACTION

Clears the current action in OEDACLI.

Syntax

```
CLEAR CURRENTACTION
```

4.3.7.2 LIST CURRENTACTION

Lists the current action in OEDACLI.

Syntax

```
LIST CURRENTACTION
```

4.3.8 DATABASE

You can add, alter, delete, or list the Oracle databases.

- [ADD DATABASE](#)
- [ALTER DATABASE](#)
- [DELETE DATABASE](#)
- [LIST DATABASES](#)

4.3.8.1 ADD DATABASE

This command adds an Oracle database; either a non-container database (non-CDB), container database (CDB), or pluggable database (PDB) to an existing CDB.

Syntax

```
ADD DATABASE
  DBNAME=database_name
  [ BLOCKSIZE=blocksize ]
  [ CHARSET=database_character_set ]
  [ DBLANG=database_language ]
  [ UNIQUENAME=database_unique_name ]
  [ DBTEMPLATE=database_template ]
  [ DBTYPE=database_type ]
  [ HOSTNAMES='host_names' ]
  [ DATADG=data_diskgroup RECO=reco_diskgroup ]
  [ VAULTLIST=exascale_vault_list EXASCALECLUSTER=exascale_cluster_name
    [ STORAGETEMPLATE = exascale_storage_template_name ] ]
WHERE
  { DBHOMEID=database_home_id |
    CLUSTERNUMBER=cluster_number DBHOMELOC=database_home_location |
    CLUSTERNAME=cluster_name DBHOMELOC=database_home_location |
    CLUSTERID=cluster_id DBHOMELOC=database_home_location |
    CLUSTERNUMBER=cluster_number CDBNAME=container_database_name |
```

```
CLUSTERNAME=cluster_name CDBNAME=container_database_name |  
CLUSTERID=cluster_id CDBNAME=container_database_name }
```

Arguments

You can specify the following attributes for the new Oracle database:

- **DBNAME** : Specifies the name of the new database.
- **BLOCKSIZE** : Optionally, specifies the block size for the new database. The default value is 8192. This argument is not required for pluggable databases.
- **CHARSET** : Optionally, specifies the character set to use for the new database. The default value is AL32UTF8. This argument is not required for pluggable databases.
- **DBLANG** : Optionally, specifies the language to use for the new database. The default value is all_langs. This argument is not required for pluggable databases.
- **UNIQUENAME** : Specifies the unique name of the new database. The default value is dbname.
- **DBTEMPLATE** : Specifies the template to use when creating the new database, ADMIN or DW. The default value is ADMIN. This argument is not required for pluggable databases.
- **DBTYPE** : Optionally, specifies the type of database to create:
 - **normal DB** : Adds a non-container database (non-CDB). This is the default value.
 - **CDB** : Adds a container database.
 - **PDB** : Adds a pluggable database to an existing CDB.
- **HOSTNAMES** : Optionally, specifies a comma-separated list of host names on which the database should run. The default value is the list of nodes registered with the database home.
- **DATADG** : Specifies the name of the DATA disk group for the new database.
This argument is required when adding a database using Oracle ASM storage. It does not apply to pluggable databases (PDBs) or databases using Exascale storage.
- **RECODG** : Specifies the name of the RECO disk group for the new database.
This argument is required when adding a database using Oracle ASM storage. It does not apply to pluggable databases (PDBs) or databases using Exascale storage.
- **VAULTLIST** : Specifies a comma-separated list of Exascale vaults used to store the database.
This argument is required when adding a database using Exascale storage. It does not apply to pluggable databases (PDBs) or databases using Oracle ASM storage.
- **EXASCALECLUSTER** : Specifies the name of the Exascale cluster that contains the Exascale vaults used to store the database.
This argument is required when adding a database using Exascale storage. It does not apply to pluggable databases (PDBs) or databases using Oracle ASM storage.
- **STORAGETEMPLATE** : Optionally, specifies the Exascale storage template applied to the Oracle Database data files.

You can use the OEDACLI `LIST STORAGETEMPLATES` command to view the list of available templates. If not specified, the default template (DATAFILE) applies.

For example, you can specify `DATA_EF_HIGHREDUNDANCY` to use the built-in Exascale template that places data files on Extreme Flash (EF) storage media (if available) using high redundancy (triple mirroring).

The following arguments are available in the `WHERE` clause:

- `DBHOMEID` : Specifies the `es.xml` ID of the database home.
- `CLUSTERNUMBER` : Specifies the cluster number in the `es.xml`, starting at 1
- `CLUSTERNAME` : Specifies the name of the cluster
- `CLUSTERID` : Specifies the `es.xml` ID of the cluster
- `DBHOMELLOCATION` : Specifies the path for the target database home.
- `CDBNAME` : Specifies the database name for the container database. This argument is required only when `DBTYPE=PDB`

Example 4-16 Adding a New Oracle Database Using OEDACLI

This example shows how to add a new database to the configuration.

1. Load the OEDA XML configuration file.

```
oedaccli> LOAD FILE NAME=Oracle-test.xml
SUCCESS - file loaded OK
Customer : ora07adm01 - Oracle
```

2. Use the command `LIST DATABASEHOMES` to identify the `databasehome_id` associated with the target database home. This database home will be used to create the new database.

```
oedaccli> LIST DATABASEHOMES
version : "CloneInstall"
cluster :
id : "c0_clusterHome"
databaseHomeName : "c0_DbHome_0"
databaseSwOwner : "c0_oracle"
databaseVersion : "12.2.0.1.170718"
databaseHomeLoc : "/u01/app/oracle/product/12.2.0.1/dbhome_1"
inventoryLocation : "/u01/app/oraInventory"
language : "all_langs"
machines :
machine : ...
...
patches :
patch :
patchNumber : "26133434"
basedir : "/u01/app/oracle"
useZfs : "false"
id : "c0_databaseHome1"
```

3. Create an action for creating the new database. You must provide a name for the new database, and the names of the DATA and RECO disk groups that the new database should use. Also, use the ID retrieved in the previous step to specify the Oracle Home in which to create this database.

```
oedaccli> ADD DATABASE DBNAME='testdb' DATADG='DATA1' RECODG='RECO1'
WHERE DBHOMEID='c0_databaseHome'
```

4. Save the action.

```
oedaccli> SAVE ACTION
```

5. Merge all actions.

```
oedaccli> MERGE ACTIONS
processMerge
processMergeActions
Merging Action : add database dbname='testdb' DATADG='DATA1'
RECODG='RECOC1'
where DBHOMEID='c0_databaseHome'
Merging ADD DATABASE
Action Validated and Merged OK
```

6. Save the action to a new Engineered Systems XML configuration file.

```
oedaccli> SAVE FILE NAME ='cli-test-2databases.xml'
File : cli-test-2databases.xml saved OK
```

7. Deploy the actions.

```
oedaccli> DEPLOY ACTIONS
Deploying Action ID : 4 add database dbname='testdb' DATADG='DATA1'
RECODG='
RECOC1' where DBHOMEID='c0_databaseHome'
Deploying ADD DATABASE
Running datapatch on database 'testdb'
Done...
Done
```

4.3.8.2 ALTER DATABASE

This command alters an existing Oracle database definition.

Syntax

```
ALTER DATABASE
{ { BLOCKSIZE=blocksize |
  CHARSET=database_character_set |
  DATADG=data_diskgroup |
  VAULT='vaultlist' [ EXASCALECLUSTER=exascaleclustername ] |
  DBLANG=database_language |
  DBNAME=database_name |
  UNIQUENAME=database_unique_name |
  DBTEMPLATE=database_template |
  DBTYPE=database_type |
  HOSTNAMES='host_names' |
  RECODG=reco_diskgroup } ... |
  { DBHOMELOC=new_dbhome_loc | DBHOMEID=new_dbhome_id } }
WHERE
{ ID=database_id |
  CLUSTERNUMBER=cluster_number DATABASENAME=database_name |
```



```
CLUSTERNAME=cluster_name DATABASENAME=database_name |
CLUSTERID=cluster_id DATABASENAME=database_name }
```

Arguments

You can modify the following attributes:

- **BLOCKSIZE:** The block size for the new database. The default value is 8192. This argument is not required for pluggable databases.
- **CHARSET:** The character set to use for the new database. The default value is AL32UTF8. This argument is not required for pluggable databases.
- **DATADG:** The name of the DATA disk group for the new database. This argument is not required for pluggable databases.
- **VAULT:** If using Exascale, specify a vault or space-separated list of two vaults for the database. Note that the value is also surrounded by single quotes (as displayed in the command syntax).
- **EXASCALECLUSTER:** Specifies the Exascale cluster that contains the specified vault or vaults. This option is required only if the vault specification cannot uniquely identify the Exascale cluster.
- **DBLANG:** The language to use for the new database. The default value is `all_langs`. This argument is not required for pluggable databases.
- **DBNAME:** The name of the new database.
- **UNIQUENAME:** The unique name of the new database. The default value is `dbname`.
- **DBTEMPLATE :** The template to use when creating the new database, `ADMIN` or `DW`. The default value is `ADMIN`. This argument is not required for pluggable databases.
- **DBTYPE:** The type of database to create:
 - `normal DB:` Specifies a non-container database (non-CDB). This is the default value.
 - `CDB:` Specifies a container database.
 - `PDB:` Specifies a pluggable database to an existing CDB.
- **HOSTNAMES:** A comma-separated list of host names on which the database should run. The default value is the list of nodes registered with the database home.
- **RECODG:** The name of the RECO disk group for the new database. This argument is not required for pluggable databases.
- **DBHOMELOC :** The location of the database home that you want to move the database to. This argument is not required for pluggable databases.
- **DBHOMEID:** The `es.xml` ID of the database home that you want to move the database to. This argument is not required for pluggable databases.

The following arguments are available in the `WHERE` clause:

- **ID:** Specifies the `es.xml` ID of the database
- **CLUSTERNUMBER:** Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **CLUSTERID:** Specifies the `es.xml` ID of the cluster
- **DATABASENAME:** Specifies the name of the database in the cluster

Usage Notes

For moving the database to use a different database home, specify only the new home location (DBHOMELoc) or the ID of the new database home (DBHOMEID).

4.3.8.3 DELETE DATABASE

This command deletes an existing Oracle database.

Syntax

```
DELETE DATABASE
WHERE
  { ID=database_id |
    CLUSTERNUMBER=cluster_number DATABASENAME=database_name |
    CLUSTERNAME=cluster_name DATABASENAME=database_name |
    CLUSTERID=cluster_id DATABASENAME=database_name }
```

Arguments

The following arguments are available in the WHERE clause:

- **ID** : Specifies the `es.xml` ID of the database
- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster
- **DATABASENAME** : Specifies the name of the database in the cluster

Example 4-17 Deleting an Oracle Database Using OEDACLI

This example shows how to delete an existing database from the configuration.

1. Load the OEDA XML configuration file.

```
oedaccli> LOAD FILE NAME=Oracle-test.xml
SUCCESS - file loaded OK
Customer : ora07adm01 - Oracle
```

2. List the databases contained in the XML configuration file, so that you can identify the database ID associated with the database you want to delete.

```
oedaccli> LIST DATABASES
version : "2"
databaseOwner : "c0_oracle"
databaseSid : "dbm01"
databaseBlockSize : "8192"
characterSet : "AL32UTF8"
databaseHome :
id : "c0_databaseHome"
machines :
machine :
.....
.....
diskGroups :
```

```

diskGroup :
id : "c0_datadg"
id : "c0_recodg"
id : "c0_dbm01"
.....
.....
databaseHome :
id : "c0_databaseHome3"
machines :
machine :
....
....
databaseTemplate : "OLTP"
databaseStyle : "ADMIN"
language : "all_langs"
diskGroups :
diskGroup :
id : "c0_datadg"
id : "c0_recodg"
id : "c0_testdb01"

```

3. Create an action to delete the target database. Use the database id identified in the previous step

```
oedacli> delete database where id='c0_testdb01'
```

4. Save the action.

```
oedacli> SAVE ACTION
```

5. Merge all actions.

```

oedacli> MERGE ACTIONS
processMerge
processMergeActions
Merging Action : delete database where id='c0_testdb01'
Merging DELETE DATABASE
Action Validated and Merged OK

```

6. Deploy the actions into the current configuration.

```

oedacli> DEPLOY ACTIONS
Deploying Action ID : 1 delete database where id='c0_testdb01'
Deploying DELETE DATABASE
Deleting database testdb
Done...
Done

```

4.3.8.4 LIST DATABASES

This command lists the existing Oracle databases.

Syntax

```
LIST DATABASES
[ WHERE
  { ID=database_id |
    CLUSTERNUMBER=cluster_number |
    CLUSTERNAME=cluster_name |
    CLUSTERID=cluster_id } ]
```

Arguments

The following arguments are available in the `WHERE` clause:

- `ID` : Specifies the `es.xml` ID of the database
- `CLUSTERNUMBER` : Specifies the cluster number in the `es.xml`, starting at 1
- `CLUSTERNAME` : Specifies the name of the cluster
- `CLUSTERID` : Specifies the `es.xml` ID of the cluster

4.3.9 DATABASEHOME

You can add, alter, delete, or list the Oracle Database homes.

- [ADD DATABASEHOME](#)
- [ALTER DATABASEHOME](#)
- [DELETE DATABASEHOME](#)
- [LIST DATABASEHOMES](#)

4.3.9.1 ADD DATABASEHOME

This command adds an Oracle database home to an existing cluster.

Syntax

```
ADD DATABASEHOME
  OWNER=owner
  DBVERSION=version
  DBHOMELOC=database_home_path
  [ DBHOMENAME=database_home_name ]
  [ INVLOC=inventory_location ]
  [ DBLANG=database_language ]
  [ MACHINELIST='machine_list' ]
  [ PATCHLIST='patch_list' ]
  [ BASEDIR=base_directory ]
  [ VOLUMESIZE=volume_size ]
  [ INSTALLTYPE={ CLONE | ZIP } ]
WHERE
  { CLUSTERNUMBER=cluster_number |
    CLUSTERNAME=cluster_name |
    CLUSTERID=cluster_id }
```

Arguments

You can specify the following attributes for the new Oracle database home:

- **OWNER** : (Mandatory) Specifies the operating system user that owns the Oracle home.
- **DBVERSION** : (Mandatory) Specifies the version of the software in the database home, including the release update (RU), for example 12.1.0.2.170418.
- **DBHOMELOC** : (Mandatory) Specifies the file system path for the new Oracle database home directory.
- **DBHOMENAME** : Specifies the name of the new database home. The default value is the standard template name, for example, `dbhome_1`.
- **INVLOC** : Specifies the location of the Oracle inventory directory. The default value is derived from the cluster home.
- **DELANG** : Specifies the language to use with the Oracle home. The default value is `all_langs`.
- **MACHINELIST** : Specifies a list of nodes to create the Oracle home on. The default value is list of nodes registered with the Grid home.
- **PATCHLIST** : Specifies a comma-separated list of patches for the database home. The default value is derived from **DBVERSION**.
- **BASEDIR** : Specifies the location of the Oracle Base directory. The default value is derived from **OWNER**.
- **VOLUMESIZE** : Specifies the size of the home directory volume. The default value is 50 GB. This argument is applicable only for database homes for Oracle VM clusters.
- **INSTALLTYPE**: Specifies the software installation method to use. The options are:
 - **CLONE**: Uses image files to clone a software installation. The image files include the base software release, and may also include additional patches and updates. By including updates in the image files, this method reduces deployment time and improves deployment consistency.

The **CLONE** software installation method is the only supported method for database homes in a virtual machine (VM) deployment. This option is also available for database homes in a physical (bare metal) deployment.
 - **ZIP**: Performs a fresh software installation by using compressed (zip) files that contain the software release. This method does not include additional patches or updates.

This option is not available for database homes in a virtual machine (VM) deployment.

If you do not specify the **INSTALLTYPE** setting, then the default setting matches the **INSTALLTYPE** setting in the containing cluster.

The following arguments are available in the **WHERE** clause:

- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster

Example 4-18 Adding a Database Home to a Deployment using OEDACLI

This example shows how to add a new Oracle home. `dbversion` should specify both the base version and the target release update (RU). In this example, we are setting up a new Oracle

```

database home with 12.1.0.2.170418. The software will be installed under /u01/app/
oracle/product/12.1.0.2/dbhome_2.

./oedaccli

oedaccli> LOAD FILE NAME=Oracle-test12.xml
SUCCESS - file loaded OK
Customer : ora07adm01 - Oracle

oedaccli> ADD DATABASEHOME OWNER='oracle' DBVERSION='12.1.0.2.170418'
DBHOMELOC='/u0
1/app/oracle/product/12.1.0.2/dbhome_2' WHERE CLUSTERNUMBER=1

oedaccli> SAVE ACTION

oedaccli> MERGE ACTIONS
processMerge
processMergeActions
Merging Action : add databasehome owner='oracle' dbversion='12.1.0.2.170418'
dbhomeloc='/u01/app/oracle/product/12.1.0.2/dbhome_2' where clusternumber=1
Merging ADD DATABASEHOME
Action Validated and Merged OK

oedaccli> DEPLOY ACTIONS
Deploying Action ID : 1 add databasehome owner='oracle'
dbversion='12.1.0.2.170418'
dbhomeloc='/u01/app/oracle/product/12.1.0.2/dbhome_2' where clusternumber=1
Deploying ADD DATABASEHOME
Checking local files..
Required files are:
SUCCESS: Required file /scratch/cpabba/Oeda/main/linux-x64/WorkDir/
p26022196_121020
_Linux-x86-64.zip exists...
SUCCESS: Required file /scratch/cpabba/Oeda/main/linux-x64/WorkDir/
p26027162_121020
_Linux-x86-64.zip exists...
SUCCESS: Required file /scratch/cpabba/Oeda/main/linux-x64/WorkDir/
p6880880_122010_
Linux-x86-64.zip exists...
SUCCESS: Required file /scratch/cpabba/Oeda/main/linux-x64/WorkDir/
linuxamd64_12102
_database_2of2.zip exists...
SUCCESS: Required file /scratch/cpabba/Oeda/main/linux-x64/WorkDir/
linuxamd64_12102
_database_1of2.zip exists...
Copying required files...
Checking status of remote files...
Checking status of existing files on remote nodes...
Getting status of local files...
Copying file: linuxamd64_12102_database_1of2.zip to node
nodeladm01.example.com
Copying file: linuxamd64_12102_database_2of2.zip to node
nodeladm01.example.com
Copying file: p26022196_121020_Linux-x86-64.zip to node
nodeladm01.example.com
Copying file: p26027162_121020_Linux-x86-64.zip to node

```

```
node1adm01.example.com
Copying file: p26022196_121020_Linux-x86-64.zip to node
node1adm02.example.com
Copying file: p26027162_121020_Linux-x86-64.zip to node
node1adm02.example.com
Completed copying files...
Extracting files...
Completed extracting files...
Validating nodes for database readiness...
Installing database software ...
Running database installer on node node1adm01.example.com ... Please wait...
After running database installer...
Patching...
Done...
Done
```

4.3.9.2 ALTER DATABASEHOME

This command alters an existing Oracle database home.

Syntax

```
ALTER DATABASEHOME
  { OWNER=owner |
    DBVERSION=version |
    DBHOMELOC=database_home_path |
    DBHOMENAME=database_home_name |
    INVLOC=inventory_location |
    DBLANG=language |
    MACHINELIST='machine_list' |
    PATCHLIST='patch_list' |
    BASEDIR=base_directory |
    INSTALLTYPE={ CLONE | ZIP } } ...
WHERE
  { ID=database_home_id |
    CLUSTERNUMBER=cluster_number DBHOMELOC=database_home_path |
    CLUSTERNAME=cluster_name DBHOMELOC=database_home_path |
    CLUSTERID=cluster_id DBHOMELOC=database_home_path }
```

Arguments

You can modify the following attributes for the Oracle database home:

- **OWNER** : Specifies the operating system user that owns the Oracle home.
- **DBVERSION** : Specifies the version of the software in the database home, including the release update (RU), for example 12.1.0.2.170418.
- **DBHOMELOC** : Specifies the file system path for the new Oracle database home directory.
- **DBHOMENAME** : Specifies the new name of the database home. The database home name must be unique within a cluster.
- **INVLOC** : Specifies the location of the Oracle inventory directory.
- **DBLANG** : Specifies the supported languages for the Oracle home.
- **MACHINELIST** : Specifies a list of nodes. The nodes must exist within the cluster.

- **PATCHLIST** : Specifies a comma-separated, custom list of patches for the database home.
- **BASEDIR** : Specifies the location of the Oracle Base directory.
- **INSTALLTYPE**: Specifies the software installation method to use. The options are:
 - **CLONE**: Uses image files to clone a software installation. The image files include the base software release, and may also include additional patches and updates. By including updates in the image files, this method reduces deployment time and improves deployment consistency.

The **CLONE** software installation method is the only supported method for database homes in a virtual machine (VM) deployment. This option is also available for database homes in a physical (bare metal) deployment.
 - **ZIP**: Performs a fresh software installation by using compressed (zip) files that contain the software release. This method does not include additional patches or updates.

This option is not available for database homes in a virtual machine (VM) deployment.

The following arguments are available in the **WHERE** clause:

- **ID** : Specifies the `es.xml` ID for the database home
- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster
- **DBHOMELOC** : Specifies the path for the database home directory.

4.3.9.3 DELETE DATABASEHOME

This command removes an existing Oracle database home.

Syntax

```
DELETE DATABASEHOME
WHERE
  { ID=database_home_id |
    CLUSTERNUMBER=cluster_number DBHOMELOC=database_home_path |
    CLUSTERNAME=cluster_name DBHOMELOC=database_home_path |
    CLUSTERID=cluster_id DBHOMELOC=database_home_path }
```

Arguments

The following arguments are available in the **WHERE** clause:

- **ID** : Specifies the `es.xml` ID for the database home
- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster
- **DBHOMELOC** : Specifies the path for the database home directory.

Usage Notes

The database home you are deleting must not contain any dependent databases.

Example 4-19 Removing an Existing Oracle Database Home

This example shows how to delete an existing Oracle Database home directory.

```
oedaccli> LOAD FILE NAME=Oracle-test12.xml
SUCCESS - file loaded OK
Customer : ora07adm01 - Oracle

oedaccli> DELETE DATABASEHOME WHERE id='c0_databaseHome1'

oedaccli> SAVE ACTION

oedaccli> MERGE ACTIONS
processMerge
processMergeActions
Merging Action : delete databasehome where id='c0_databaseHome1'
Merging DELETE DATABASEHOME
Action Validated and Merged OK

oedaccli> DEPLOY ACTIONS
Deploying Action ID : 3 delete databasehome where id='c0_databaseHome1'
Deploying DELETE DATABASEHOME
Validating Oracle home..
Deinstalling database home c0_DbHome_1
Unmounting file system..
Updating /etc/fstab entries...
Done...
Done
```

4.3.9.4 LIST DATABASEHOMES

This command lists the existing Oracle database homes.

Syntax

```
LIST DATABASEHOMES
[ WHERE
    { ID=database_home_id |
      CLUSTERNUMBER=cluster_number [ NAME=database_home_name ] |
      CLUSTERNAME=cluster_name [ NAME=database_home_name ] |
      CLUSTERID=cluster_id [ NAME=database_home_name ] } ]
```

Arguments

The following arguments are available in the `WHERE` clause:

- **ID** : Specifies the `es.xml` ID for the target database home
- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster
- **NAME** : Specifies the name of the target database home

4.3.10 DISKGROUP

You can add, alter, delete, or list the Oracle ASM disk groups.

- [ADD DISKGROUP](#)
- [ALTER DISKGROUP](#)
- [ALTER DISKGROUPS](#)
- [DELETE DISKGROUP](#)
- [LIST DISKGROUPS](#)

4.3.10.1 ADD DISKGROUP

This command adds an Oracle ASM disk group.

Syntax

```
ADD DISKGROUP
  [ ACFSNAME=acfs_name ]
  [ ACFSPATH=acfs_path ]
  [ ACFSSIZE=acfs_size ]
  [ CELLLIST='cell_list' ]
  DISKGROUPNAME=diskgroup_name
  [ DISKGROUPLOCATION=diskgroup_location ]
  DISKGROUPSIZE=diskgroup_size
  [ OCRVOTE=ocr_voting_disks_included ]
  [ QUORUMDISK=quorum_disks_included ]
  REDUNDANCY=redundancy_level
  SLICESIZE=slice_size
  [ SPARSE=sparse ]
  [ TYPE=diskgroup_type [ DATABASENAME=database_name ] ]
WHERE
  { CLUSTERNUMBER=cluster_number |
    CLUSTERNAME=cluster_name |
    CLUSTERID=cluster_id }
```

Arguments

You can specify the following attributes for the new Oracle ASM disk group:

- **ACFSNAME** : Specifies the name for the Oracle ACFS volume on this disk group
- **ACFSPATH** : Specifies the path for the Oracle ACFS file system
- **ACFSSIZE** : Specifies the size for the Oracle ACFS volume on this diskgroup
- **CELLLIST** : A comma separated list of cells for this disk group, enclosed in single quotes
- **DISKGROUPNAME** : (Mandatory) Specifies the name of the disk group
- **DISKGROUPLOCATION** : Specifies the media type where the disk group is located. The valid values are **FLASHDISK** or **HARDDISK**. The default value is **HARDDISK**.

The same disk group location is recommended for all non-RECO disk groups across all clusters in the Exadata configuration file (*es.xml*). If the RECO disk group location is **FLASHDISK**, then **FLASHDISK** is the recommended location for all disk groups.

- **DISKGROUPSIZE** : (Mandatory) Specifies the size of the disk group
- **OCRVOTE** : A boolean variable, where **TRUE** indicates the disk group should store the OCR and voting disks for the cluster.
- **QUORUMDISK** : A boolean variable, where **TRUE** indicates the disk group requires a quorum
- **REDUNDANCY** : (Mandatory) Specifies the redundancy level for the disk group. The valid values are **NORMAL** or **HIGH**.
- **SLICESIZE** : (Mandatory) Specifies the slice size on disk for each grid disk for this disk group.
- **SPARSE** : A boolean variable, where **TRUE** indicates this should be sparse disk group.
- **TYPE** : Specifies the disk group type, which can be **DATA**, **RECO**, **DBFS** or **OTHER**. The default value is **OTHER**. If you want the database to use this disk group, then the type must be set to **DATA** or **RECO**.
- **DATABASENAME** : Specifies the name of the target database for disk groups of type **DATA** or **RECO**.

The following arguments are available in the **WHERE** clause:

- **CLUSTERNUMBER** : Specifies the cluster number in the **es.xml**, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the **es.xml** ID of the cluster

Example 4-20 Adding a Disk Group

The following example shows an OEDACLI session that uses the **ADD DISKGROUP** command along with typical output at each step.

```
oedaccli> add diskgroup diskgroupname=OTHER7 diskgroupsize=1TB
redundancy=HIGH
celllist='dbm01celadm07.example.com,dbm01celadm08.example.com,dbm01celadm09.example.com' where clustername=Cluster-c1
oedaccli> save action
oedaccli> merge actions
processMerge
processMergeActions
Merging Action : add diskgroup diskgroupname=OTHER7 diskgroupsize=1TB
redundancy=HIGH
celllist='dbm01celadm07.example.com,dbm01celadm08.example.com,dbm01celadm09.example.com' where clustername=Cluster-c1
Merging ADD DISKGROUP
Action Validated and Merged OK
oedaccli> deploy actions
Deploying Action ID : 3 add diskgroup diskgroupname=OTHER7
diskgroupsize=1TB redundancy=HIGH
celllist='dbm01celadm07.example.com,dbm01celadm08.example.com,dbm01celadm09.example.com' where clustername=Cluster-c1
Deploying ADD DISKGROUP
Diskgroup OTHER7 will be created on Storage Servers
[dbm01celadm07.example.com, dbm01celadm08.example.com,
dbm01celadm09.example.com]
Validating free space....
Creating Grid Disks for ASM Disk Group OTHER7
Creating ASM Disk Group OTHER7
```

```

Updating ASM Diskstring...
Checking ASM Disk Group status...
Completed creation of ASM Disk Group OTHER7
Done...
Done

```

4.3.10.2 ALTER DISKGROUP

This command alters attributes for an existing Oracle ASM disk group.

Syntax

```

ALTER DISKGROUP
  { ACFSNAME=acfs_name |
    ACFSPATH=acfs_path |
    ACFSSIZE=acfs_size |
    CELLLIST='cell_list' |
    DISKGROUPNAME=diskgroup_name |
    DISKGROUPLOCATION=diskgroup_location |
    DISKGROUPSIZE=diskgroup_size |
    OCRVOTE=ocr_voting_disks_included |
    QUORUMDISK=quorum_disks_included |
    REDUNDANCY=redundancy_level |
    SLICESIZE=slice_size |
    SPARSE=sparse } ...
WHERE
  { ID=diskgroup_id |
    CLUSTERNUMBER=cluster_number DISKGROUPNAME=diskgroup_name |
    CLUSTERNAME=cluster_name DISKGROUPNAME=diskgroup_name |
    CLUSTERID=cluster_id DISKGROUPNAME=diskgroup_name }

```

Arguments

You can modify the following attributes of an Oracle ASM disk group:

- **ACFSNAME** : Specifies a new name for the Oracle ACFS volume on this disk group
- **ACFSPATH** : Specifies the new path for the Oracle ACFS file system
- **ACFSSIZE** : Specifies a new size for the Oracle ACFS volume on this disk group
- **CELLLIST** : A comma separated list of cells for this disk group, enclosed in single quotes
- **DISKGROUPNAME** : Specifies a new name for the disk group
- **DISKGROUPLOCATION** : Specifies the media type where the disk group is located. The valid values are **FLASHDISK** or **HARDDISK**. The default value is **HARDDISK**.

The same disk group location is recommended for all non-RECO disk groups across all clusters in the Exadata configuration file (*es.xml*). If the RECO disk group location is **FLASHDISK**, then **FLASHDISK** is the recommended location for all disk groups.

- **DISKGROUPSIZE** : Specifies a new size for the disk group
- **OCRVOTE** : A boolean variable, where **TRUE** indicates the disk group should store the OCR and voting disks for the cluster.
- **QUORUMDISK** : A boolean variable, where **TRUE** indicates the disk group requires a quorum

- **REDUNDANCY** : Specifies the redundancy level for the disk group. The valid values are **NORMAL** or **HIGH**.
- **SLICESIZE** : Specifies the slice size on disk for each grid disk for this disk group.
- **SPARSE** : A boolean variable, where **TRUE** indicates this should be sparse disk group.

The following arguments are available in the **WHERE** clause:

- **ID** : Specifies the `es.xml` ID for the disk group
- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster
- **DISKGROUPNAME** : Specifies the name of the disk group

Usage Notes

- With the **ALTER DISKGROUP** command, you can only deploy the following modifications to an existing disk group:
 - You can alter the size of an existing disk group by setting the **DISKGROUPSIZE** attribute.
 - You can change the storage servers that contain the disk group by setting the **CELLLIST** attribute. However, if you remove a storage server from a disk group, you must ensure that the remaining storage servers have enough space to accommodate the existing data.
- Changes to all other disk group attributes may only be saved to the OEDA XML configuration file (`es.xml`). Consequently, you cannot use the **ALTER DISKGROUP** command in conjunction with the **DEPLOY ACTIONS** command to change the name of an existing disk group, for example.
- If you modify the redundancy, the volume size will be maintained, however the slice size will be recalculated.

Example 4-21 Resizing an Oracle ASM Disk Group Using OEDACLI

This example shows how to resize an existing Oracle ASM disk group using OEDACLI.

1. Load the OEDA XML configuration file.

```
oedaccli> LOAD FILE NAME=Oracle-test.xml
SUCCESS - file loaded OK
Customer : ora07adm01 - Oracle
```

2. Use the command **LIST DISKGROUPS** to view the disk group details.

```
oedaccli> LIST DISKGROUPS
version : "PhysicalLayoutLegacy"
diskGroupName : "DBFS_DG"
redundancy : "NORMAL"
sliceSize : "default"
machines :
machine :
...
...
cellDisks : "ALL"
diskGroupSize : "436G"
```

```
ocrVote : "true"
quorumDisk : "false"
id : "c0_dbfsdg"
version : "PhysicalLayoutLegacy"
diskGroupName : "DATA1"
redundancy : "NORMAL"
sliceSize : "2953G"
machines :
..
..
cellDisks : "ALL"
diskGroupSize : "53154G"
ocrVote : "false"
quorumDisk : "false"
id : "c0_datadg"
version : "PhysicalLayoutLegacy"
diskGroupName : "RECOC1"
redundancy : "NORMAL"
sliceSize : "ALL"
machines :
..
..
cellDisks : "ALL"
diskGroupSize : "13284G"
ocrVote : "false"
quorumDisk : "false"
id : "c0_recodg"
```

3. Create an action for resizing the disk group.

```
oedaccli> ALTER DISKGROUP DISKGROUPSIZE='2500G' WHERE CLUSTERNUMBER=1
DISKGROUPN
AME='DATA1'
```

4. Save the action.

```
oedaccli> SAVE ACTION
```

5. Merge all actions.

```
oedaccli> MERGE ACTIONS
processMerge
processMergeActions
Merging Action : alter diskgroup diskgroupsize='2500G' where
diskgroupname='DAT
AC1' clusternumber=1
Merging ALTER DISKGROUP
Action Validated and Merged OK
```

6. Deploy the actions.

```
oedaccli> DEPLOY ACTIONS
Deploying Action ID : 3 alter diskgroup diskgroupsize='2500G' where
diskgroupna
me='DATA1'
```

```

Deploying ALTER DISKGROUP
Deploying Action ID : 4 alter diskgroup diskgroupsize='2500G' where
diskgroupna
me='DATA1' clusternumber=1
Deploying ALTER DISKGROUP
Validating ASM Disk status..
Disks in ASM Disk Group DATA1 have valid status
Shrinking Disk Group size to 2500 GB
Resizing ASM Disk Group..
Checking Reblance operations in disk group DATA1
Rebalance operations completed in disk group DATA1
Original Grid Disk Size 2953 GB, new size 138 GB on
node1celadm02.example.com
Original Grid Disk Size 2953 GB, new size 138 GB on
node1celadm03.example.com
Original Grid Disk Size 2953 GB, new size 138 GB on
node1celadm01.example.com
Updated/New Disk Group size 2484 GB, original size 53154 GB.
Done...
Done

```

4.3.10.3 ALTER DISKGROUPS

This command alters disk group sizing attributes for existing Oracle ASM disk groups.

Syntax

```

ALTER DISKGROUPS
  DATASPLIT=data_split_percent
  RECOSPLIT=reco_split_percent
  DATAREDUNDANCY=data_redundancy_level
  RECOREDUNDANCY=reco_redundancy_level
  SPLITDISK=percent_of_cell_disk
WHERE
  DATADG=data_diskgroup
  RECODG=reco_diskgroup
  { CLUSTERNAME=cluster_name |
    CLUSTERNUMBER=cluster_number |
    CLUSTERID=cluster_id }

```

Arguments

You can modify the following sizing attributes of the DATA and RECO Oracle ASM disk groups:

- **DATASPLIT** : Specifies a percentage of the Oracle ASM volume allocated to the DATA disk group. The sum of DATASPLIT and RECOSPLIT cannot exceed 100.
- **RECOSPLIT** : Specifies a percentage of the Oracle ASM volume allocated to the RECO disk group. The sum of DATASPLIT and RECOSPLIT cannot exceed 100.
- **DATAREDUNDANCY** : Specifies the redundancy level for the DATA disk group. The accepted values are NORMAL or HIGH.
- **RECOREDUNDANCY** : Specifies the redundancy level for the RECO disk group. The accepted values are NORMAL or HIGH.

- **SPLITDISK** : Specifies the percentage of the cell disk to allocate to the DATA and RECO disk groups. The specified value must be an integer between 1 and 100.

The following arguments are available in the `WHERE` clause:

- **DATADG** : Specifies the name of the DATA disk group in the cluster
- **RECOG** : Specifies the name of the RECO disk group in the cluster
- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster

4.3.10.4 DELETE DISKGROUP

This command deletes an existing Oracle ASM disk group.

Syntax

```
DELETE DISKGROUP
WHERE
  { ID=diskgroup_id |
    CLUSTERNUMBER=cluster_number DISKGROUPNAME=diskgroup_name |
    CLUSTERNAME=cluster_name DISKGROUPNAME=diskgroup_name |
    CLUSTERID=cluster_id DISKGROUPNAME=diskgroup_name }
```

Arguments

The following arguments are available in the `WHERE` clause:

- **ID** : Specifies the `es.xml` ID for the disk group
- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster
- **DISKGROUPNAME** : Specifies the name of the disk group

Example 4-22 Deleting a Disk Group

The following example shows an OEDACLI session that uses the `DELETE DISKGROUP` command along with typical output at each step.

```
oedacli> delete diskgroup where id=10dff870-b824-cdb7-0542-8a03a88e9653
oedacli> save action
oedacli> merge actions
processMerge
processMergeActions
Merging Action : delete diskgroup where id=10dff870-b824-
cdb7-0542-8a03a88e9653
Merging DELETE DISKGROUP
Action Validated and Merged OK
oedacli> deploy actions
Deploying Action ID : 4 delete diskgroup where id=10dff870-b824-
cdb7-0542-8a03a88e9653
Deploying DELETE DISKGROUP
```



```

Dropping ASM Disk Group DATA1
ASM Disk Group DATA1 does not exist..
Grid Disks for Disk Group DATA1 exist on cells dbm01celadm07.example.com
Grid Disks for Disk Group DATA1 exist on cells dbm01celadm08.example.com
Grid Disks for Disk Group DATA1 exist on cells dbm01celadm09.example.com
Dropping Grid Disks on [dbm01celadm07.example.com,
dbm01celadm08.example.com, dbm01celadm09.example.com]
Deleting quorum devices...
Completed dropping ASM Disk Group DATA1
Done...
Done

```

4.3.10.5 LIST DISKGROUPS

This command lists the existing Oracle ASM disk groups.

Syntax

```

LIST DISKGROUPS
[ WHERE
    { ID=diskgroup_id |
      CLUSTERNUMBER=cluster_number |
      CLUSTERNAME=cluster_name |
      CLUSTERID=cluster_id } ]

```

Arguments

The following arguments are available in the `WHERE` clause:

- `ID` : Specifies the `es.xml` ID for the disk group
- `CLUSTERNUMBER` : Specifies the cluster number in the `es.xml`, starting at 1
- `CLUSTERNAME` : Specifies the name of the cluster
- `CLUSTERID` : Specifies the `es.xml` ID of the cluster

4.3.11 DOM0

You can list Oracle VM management domains (dom0s).

- [LIST DOM0S](#)

4.3.11.1 LIST DOM0S

This command lists Oracle VM management domains (dom0s).

Syntax

```

LIST DOM0S

```

Usage Notes

This command is deprecated. Instead, use:

```
LIST MACHINES WHERE TYPE=DOM0
```

4.3.12 DOMAIN

You can get live information from a domain.

- [GET DOMAIN](#)

4.3.12.1 GET DOMAIN

This command displays live domain information from a running system.

Syntax

```
GET DOMAIN
  ACTION=action
WHERE
  { ID=id |
    HOSTNAME=hostname |
    CLUSTERNUMBER=cluster_number { COMPUTENUMBER=compute_number |
    STORAGENUMBER=storage_number } |
    CLUSTERNAME=cluster_name { COMPUTENUMBER=compute_number |
    STORAGENUMBER=storage_number } |
    CLUSTERID=cluster_id { COMPUTENUMBER=compute_number |
    STORAGENUMBER=storage_number } }
```

Arguments

ACTION : Specifies the information that the command returns.

If the action is applied to a guest, then the command displays specific information for the guest.

If the action is applied to the host, then the command displays information for the hypervisor.

- **LISTIMAGES**: Displays image file details.
- **LISTDOMAIN**: Displays domain details.
- **LISTMEMORY**: Displays memory details.
- **LISTVCPU**: Displays CPU details.

The following arguments are available in the **WHERE** clause:

- **ID** : Specifies the `es.xml` ID for the target machine
- **HOSTNAME** : Specifies the current hostname of the target machine
- **CLUSTERNUMBER** : Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERID** : Specifies the `es.xml` ID of the cluster
- **COMPUTENUMBER** : Specifies the `es.xml` compute node number in the cluster, starting at 1

- **STORAGENUMBER** : Specifies the `es.xml` storage server number in the cluster, starting at 1

4.3.13 ES

You can discover an engineered system or alter a limited number of attributes of an engineered system.

- **ALTER ES**
This command alters the attributes of an engineered system.
- **DISCOVER ES**
This command discovers an existing configuration of an engineered system.
- **VALIDATE ES**

4.3.13.1 ALTER ES

This command alters the attributes of an engineered system.

Syntax

```
ALTER ES
  PAAS = 'pass_mode' |
  CUSTOMERNAME = customer_name |
  CUSTOMERDEPT = customer_dept
```

Arguments

You can modify the following attributes for the engineered system:

- **PAAS** : A boolean value, `TRUE` or `FALSE`, which specifies whether to enable PaaS mode.
- **CUSTOMERNAME** : Specifies the customer name.
- **CUSTOMERDEPT** : Specifies the customer department.

4.3.13.2 DISCOVER ES

This command discovers an existing configuration of an engineered system.

Syntax

```
DISCOVER ES
  HOSTNAMES = host_names
  LOCATION = directory_name
```

Arguments

You can use the following arguments when discovering an engineered system configuration:

- **HOSTNAMES**: Specifies the list of nodes to be discovered, with each host name separated by a ',' or a space. The list must be enclosed in quotation marks, for example, `'node1,node2'` or `'node1 node2'`. For virtual environments with user domains, you need to provide a list of only management domain (dom0) host names and storage servers.
- **LOCATION**: Specifies the target directory for file creation.

Usage Notes

The following are requirements and limitations of the `DISCOVER ES` command:

- Supported only on Exadata. ZDLRA is not supported.
- All hosts must have the same root password.
- Discover will not work with custom Operating System profiles and customized SQL*Plus prompts.
- Only RAC databases are discovered. Single instance databases and SIHA databases are not discovered.
- Oracle Clusterware must be running.
- Only online databases are discovered.
- If `DISCOVER ES` cannot log into databases as SYSDBA using Operating System authentication databases are skipped.
- Only database resources registered with Oracle Clusterware are discovered.
- Virtual machines that are not running are not discovered.

4.3.13.3 VALIDATE ES

Validates certain aspects of an OEDA XML configuration file (`es.xml`).

Syntax

```
VALIDATE ES
```

4.3.14 EXASCALE

- [DEPLOY EXASCALE](#)

This command performs Exascale deployment actions.

4.3.14.1 DEPLOY EXASCALE

This command performs Exascale deployment actions.

Syntax

```
DEPLOY EXASCALE
  ACTION = action
WHERE
  { CLUSTERNUMBER = clusternumber [ DATABASENAME = databasename ] |
    CLUSTERNAME = clustername [ DATABASENAME = databasename ] |
    CLUSTERID = clusterid [ DATABASENAME = databasename ] |
    DATABASENAME = databasename }
```

Arguments

- **ACTION:** Specifies the Exascale deployment action, which is one of the following:
CONFIGURECELLS, CONFIGURECOMPUTES, CREATEDBWALLET, CREATEGIVault, CREATEDBVault,

UNDOCONFIGURECELLS, UNDOCONFIGURECOMPUTES, UNDOCREATEDBWALLET, UNDOCREATEGIVALT,
UNDOCREATEDBVAULT.

The following arguments are available in the `WHERE` clause:

- `CLUSTERNUMBER` : Specifies the cluster number in the OEDA configuration file (`es.xml`), starting at 1.
- `CLUSTERNAME` : Specifies the name of the cluster.
- `CLUSTERID` : Specifies the OEDA configuration file (`es.xml`) ID of the cluster.
- `DATABASENAME` : Specifies the name of the database.

Usage Notes

- The `CONFIGURECELLS` action must be performed before the `CONFIGURECOMPUTES` action.
- The `CONFIGURECELLS` and `CONFIGURECOMPUTES` actions must be performed before any wallet or vault actions.
- The `CONFIGURECOMPUTES` action creates an Exascale wallet for the Oracle Grid Infrastructure operating system (OS) owner. This wallet also supports Oracle Database in a non-role-separated Exadata configuration, where the Oracle Grid Infrastructure OS owner is the same as the Oracle Database OS owner.
- The `CREATEDBWALLET` action only creates an Exascale wallet for an Oracle Database OS owner in a role-separated Exadata configuration, where the Oracle Database OS owner is not the same as the Oracle Grid Infrastructure OS owner.
- Undo actions must be performed in the following order: `UNDOCREATEDBVAULT`, `UNDOCREATEGIVALT`, `UNDOCREATEDBWALLET`, `UNDOCONFIGURECOMPUTES` and `UNDOCONFIGURECELLS`.

4.3.15 EXASCALECLUSTER

- **ADD EXASCALECLUSTER**
This command adds a new Exascale cluster.
- **ALTER EXASCALECLUSTER**
This command alters an existing Exascale cluster.
- **DELETE EXASCALECLUSTER**
This command deletes an existing Exascale cluster.
- **LIST EXASCALECLUSTER**
This command displays details for an existing Exascale cluster.
- **LIST EXASCALECLUSTERS**
This command displays details for all Exascale clusters.

4.3.15.1 ADD EXASCALECLUSTER

This command adds a new Exascale cluster.

Syntax

```
ADD EXASCALECLUSTER
  NAME = exascaleclustername
```

```
[ VIP = ersvip  
  IP = ersip ]
```

Arguments

- **NAME:** Specifies the name of the Exascale cluster.
- **VIP:** Specifies the DNS name for the virtual IP (VIP) interface that provides access to Exascale control services (also known as Exascale RESTful Services or ERS).
- **IP:** Specifies the IP address for the ERS VIP.

Usage Notes

- The network interface specified by **VIP** and **IP** is associated with a highly-available virtual IP (VIP) network interface, which provides a consistent network end point for Exascale control services. The VIP is hosted by one of the Exadata storage servers, which also runs a front-end ERS instance. If the storage server or ERS instance becomes unavailable, then the VIP moves to another server hosting a front-end ERS instance.

You should define the ERS VIP if possible. The option to define an Exascale cluster without an ERS VIP primarily exists to support Exascale configurations on existing Exadata systems that have no spare IP addresses to support the ERS VIP.

4.3.15.2 ALTER EXASCALECLUSTER

This command alters an existing Exascale cluster.

Syntax

```
ALTER EXASCALECLUSTER  
  [ NAME = newexascaleclustername ]  
  [ VIP = newersvip ]  
  [ IP = newersip ]  
WHERE  
  NAME = exascaleclustername
```

Arguments

- **NAME:** Changes the name of the Exascale cluster.
- **VIP:** Specifies the DNS name for the virtual IP (VIP) interface that provides access to Exascale control services (also known as Exascale RESTful Services or ERS).
- **IP:** Specifies the IP address for the ERS VIP.

The following argument is available in the **WHERE** clause:

- **NAME:** Specifies the name for the Exascale cluster that you want to alter.

Usage Notes

- The network interface specified by **VIP** and **IP** is associated with a highly-available virtual IP (VIP) network interface, which provides a consistent network end point for Exascale control services. The VIP is hosted by one of the Exadata storage servers, which also runs a front-end ERS instance. If the storage server or ERS instance becomes unavailable, then the VIP moves to another server hosting a front-end ERS instance.

4.3.15.3 DELETE EXASCALECLUSTER

This command deletes an existing Exascale cluster.

Syntax

```
DELETE EXASCALECLUSTER
WHERE
    NAME = exascaleclustername
```

Arguments

The following argument is available in the `WHERE` clause:

- **NAME:** Specifies the name for the Exascale cluster that you want to delete.

Usage Notes

- You cannot delete an Exascale cluster while it provides resources to Oracle Grid Infrastructure or an Oracle database.

4.3.15.4 LIST EXASCALECLUSTER

This command displays details for an existing Exascale cluster.

Syntax

```
LIST EXASCALECLUSTER
WHERE
    NAME = exascaleclustername
```

Arguments

The following argument is available in the `WHERE` clause:

- **NAME:** Specifies the name for the Exascale cluster that you want to display.

4.3.15.5 LIST EXASCALECLUSTERS

This command displays details for all Exascale clusters.

Syntax

```
LIST EXASCALECLUSTERS
```

4.3.16 EXITONERROR

You can set the exit status for OEDACLI.

- **SET EXITONERROR**
This command controls the exit status for OEDACLI when an error is encountered.

4.3.16.1 SET EXITONERROR

This command controls the exit status for OEDACLI when an error is encountered.

Syntax

```
SET EXITONERROR ENABLE={ true | false }
```

Arguments

- **ENABLE:** When `ENABLE=true`, a run-time error causes OEDACLI to terminate with exit code 1. This option may be useful when OEDACLI is used inside a script.

4.3.17 FILE

You can manage the Engineered System XML file (`es.xml`).

- [LOAD FILE](#)
- [MERGE FILES](#)
- [SAVE FILE](#)
- [SAVE FILES](#)

4.3.17.1 LOAD FILE

Use this command to load an Engineered System XML file (`es.xml`) into OEDACLI memory for further use.

Syntax

```
LOAD FILE NAME=file_name
```

Arguments

- **NAME :** Specifies the name of an existing Engineered System XML file

Example 4-23 Loading a File into OEDACLI for Editing

```
oedacli> LOAD FILE NAME=Oracle-test12.xml
SUCCESS - file loaded OK
Customer : testadm01 - Oracle
```

4.3.17.2 MERGE FILES

Use this command to merge multiple Engineered System XML files into a new file.

Syntax

```
MERGE FILES
  INPUT='input_files'
  OUTPUT=output_file
```


Arguments

- **INPUT:** Specifies a space-separated list of Engineered System XML files to merge. The list of input files is surrounded by single quotes ('').
- **OUTPUT:** Specifies the new target Engineered System XML file. The file is overwritten if it already exists.

4.3.17.3 SAVE FILE

Use this command to save a current set of actions to an Engineered System XML file.

Syntax

```
SAVE FILE
  [ NAME=file_name ]
  [ MAC={ true | false } ]
```

Arguments

- **NAME:** Optionally specifies the name of a new Engineered System XML file. If you do not specify the **NAME** argument, then the command overwrites the source file.
- **MAC:** When **MAC=true**, OEDACLI updates the Engineered System XML file with MAC addresses for the system components. The ILOM interfaces must be available for this operation.

4.3.17.4 SAVE FILES

Use this command to generate a full set of OEDA configuration files.

Syntax

```
SAVE FILES
  LOCATION=dir_name
  [ MAC={ true | false } ]
```

Arguments

- **LOCATION:** Specifies the name of a target directory for file creation.
- **MAC:** When **MAC=true**, OEDACLI updates the Engineered System XML file with MAC addresses for the system components. The ILOM interfaces must be available for this operation.

Usage Notes

The specified **LOCATION** directory is created if it does not already exist.

4.3.18 GUEST

You can clone or delete an Oracle VM guest.

- **CLONE GUEST**
This command clones an existing VM guest and creates a new fully functional guest with storage, users, and Oracle RAC instances.
- **CREATE GUEST**
This command creates a new virtual machine (VM) guest.
- **DELETE GUEST**
This command deletes an existing Oracle VM guest, and removes all of its configuration: instances, software installations, storage, and users from the cluster.

4.3.18.1 CLONE GUEST

This command clones an existing VM guest and creates a new fully functional guest with storage, users, and Oracle RAC instances.

Prerequisites

Before you can clone a virtual machine (VM) guest, you must download the appropriate software images from My Oracle Support and extract the files. Place the extracted "klone.zip" files for Oracle Grid Infrastructure and Oracle Database in the `/EXAVMIMAGES/onecommand/<platform>/WorkDir` directory in the VM host where the new guest clone will be created.

Syntax

```
CLONE GUEST
  { SRCNAME=source_guest TGTNAME=target_guest |
    TGTNAMES=target_guests }
[ WHERE
  STEPNAME=stepname ]

SET PARENT NAME=parent_name
[ SET ADMINNET NAME=admin_name IP=admin_ip [ GATEWAY=gateway
NETMASK=netmask ] ]
[ SET BACKUPNET NAME=backup_name IP=backup_ip ]
SET PRIVNET NAME1=priv_name_1 IP1=priv_ip1 NAME2=priv_name_2 IP2=priv_ip2
[ SET INTERCONNECT NAME1=priv_name_1 IP1=priv_ip1 NAME2=priv_name_2
IP2=priv_ip2 ]
SET CLIENTNET NAME=client_name IP=client_ip [ NATHOSTNAME=nat_host_name
NATIP=nat_ip NATDOMAINNAME=nat_domain_name, NATNETMASK=nat_netmask ]
SET VIPNET NAME=vip_name,IP=vip_ip
[ SET VCPU COUNT=guest_cpu ]
[ SET VMEM SIZE=guest_memory ]
```

Arguments

- **SRCNAME:** Specifies the name of an existing guest VM from which the new guest VM is derived.
- **TGTNAME:** Specifies the host name of the new guest VM that you are creating with the clone operation.
- **TGTNAMES:** Specifies a comma-separated list of new guest VMs to add.
- **STEPNAME:** Specifies the deployment step to perform, which is one of the following:
 - `CREATE_GUEST`

- CREATE_USERS
- CELL_CONNECTIVITY
- CONFIG_CLUSTERWARE
- RUN_ROOTSCRIPT
- EXTEND_DBHOME
- ADD_INSTANCE

You can specify the following additional settings when specifying `CLONE GUEST` with the `WHERE STEPNAME=CREATE_GUEST` clause or when specifying `CLONE GUEST` without using any `WHERE STEPNAME` clause:

- **PARENT**: Specifies the name of the virtual machine (VM) host (Oracle Linux KVM or Oracle VM) that hosts the new guest.
- **ADMINNET** : Specifies the DNS name and IP address to use for the administration network. Also, optionally specifies the gateway and netmask for the administration network.
- **BACKUPNET** : Specifies the DNS name and IP address to use for the backup network. The DNS name may optionally include the domain name. Not required on guests without a backup network.
- **PRIVNET** : Specifies the names and IP addresses to use for the first and second private networks.
- **INTERCONNECT**: Only required if the compute nodes are configured with Exadata Secure RDMA Fabric Isolation or InfiniBand security (PKEYS). Specifies the names and IP addresses to use for the first and second private networks.
- **CLIENTNET** : Specifies the DNS name and IP address to use for the client network. May optionally include host name, IP address, domain name, and netmask details to support Network Address Translation (NAT).
- **VIPNET** : Specifies the DNS name and IP address to use for the virtual IP (VIP) network.

Usage Notes

- For configurations with Exadata Secure RDMA Fabric Isolation or InfiniBand security (PKEYS), **INTERCONNECT** defines the private network that connects the database nodes and **PRIVNET** defines the private storage network.
- When you load the XML configuration file to use when cloning a guest domain, the XML file should have the management domain (dom0) node of the source VM defined but not allocated to the cluster that is being extended into the guest domain.
- You should save the modified XML file after the actions are merged so that you have a new XML file that reflects the addition of the new guest domain.
- You can use the `CLONE GUEST` command to add a VM to a server that already contains a VM in the same cluster. Consequently, you can create a configuration where one VM host contains multiple VMs that are members of the same cluster. However, this capability is only available on systems not configured with Exadata Secure RDMA Fabric Isolation.
- During deployment, the cloned guest inherits various configuration attributes from the source guest, including the client network configuration and the backup network configurations (if present).

If all of the KVM hosts have the same network configuration, then the inherited attributes work as expected.

However, if the new KVM host uses a different physical network configuration, deployment of the cloned guest will fail. This situation is most likely when an Exadata system contains different versions of compute node hardware. For example, when adding an X10M server to an X8M-2 rack.

In this case, you must manually adjust the relevant network definition by using the `ALTER NETWORK` command before deployment. Contact Oracle Support for details.

Example 4-24 Cloning a Guest VM

This example shows how to clone a new guest from an existing node in the cluster. In this example, the source node name is `exa01adm01vm01.example.com` and the new node is `exa01adm03vm01.example.com`. A series of `SET` commands are used to specify the configuration for the new guest VM.

The XML file loaded at the beginning of this example has the `dom0` node (`host01adm01.example.com`) defined but not allocated to the cluster that is being extended into the guest.

```
oedaccli> LOAD FILE NAME=exa01-pre-cloning-node3.xml
oedaccli> CLONE GUEST SRCNAME='exa01adm01vm01.example.com'
TGTNAME='exa01adm03vm01.example.com'
oedaccli> SET PARENT NAME='exa01adm03.example.com'
oedaccli> SET ADMINNET NAME='exa01adm03vm01.example.com' IP='10.xxx.xx.x'
oedaccli> SET CLIENTNET NAME='exa01client03vm01.example.com' IP='10.xxx.xx.x'
oedaccli> SET PRIVNET NAME1='exa01adm03vm01-priv' IP1='192.168.16.8'
NAME2='exa01adm01vm03-priv2' IP2='192.168.16.9'
oedaccli> SET VIPNET NAME='exa01client03m01-vip.example.com' IP='10.xxx.xx.x'
oedaccli> SAVE ACTION
oedaccli> MERGE ACTIONS
oedaccli> SAVE FILE NAME='exa01-cloned-node3-rac.xml'
oedaccli> DEPLOY ACTIONS
```

Example 4-25 Cloning Multiple Guests Using TGTNAMES

This example clones `dbguest23` and creates the new guests `dbguest24` and `dbguest25`. By using the `TGTNAMES` argument, both new guests are deployed at the same time in the final `DEPLOY ACTIONS`.

```
CLONE GUEST SRCNAME='dbguest23' TGTNAME='dbguest24'
SET ADMINNET name='dbguest24' ip='10.128.161.202'
...
SAVE ACTION FORCE

CLONE GUEST SRCNAME='dbguest23' TGTNAME='dbguest25'
SET ADMINNET name='dbguest25' ip='10.128.161.203'
...
SAVE ACTION FORCE

CLONE GUEST TGTNAMES='dbguest24,dbguest25'

SAVE ACTION
MERGE ACTIONS
DEPLOY ACTIONS
```

Related Topics

- Expanding an Oracle VM RAC Cluster on Exadata Using OEDACLI
- Expanding an Oracle RAC Cluster in Oracle Linux KVM Using OEDACLI

4.3.18.2 CREATE GUEST

This command creates a new virtual machine (VM) guest.

Syntax

```
CREATE GUEST
    NAME=guest_name

SET PARENT NAME=parent_name
SET ADMINNET NAME=admin_name [ IP=admin_ip GATEWAY=gateway NETMASK=netmask
VLANID=vlan_ID ]
SET CLIENTNET NAME=client_name [ IP=client_ip GATEWAY=gateway NETMASK=netmask
VLANID=vlan_ID ]
[ SET CLIENTNET NATHOSTNAME=nat_host_name NATIP=nat_ip
NATDOMAINNAME=nat_domain_name, NATNETMASK=nat_netmask ]
[ SET BACKUPNET NAME=backup_name [ IP=backup_ip GATEWAY=gateway
NETMASK=netmask VLANID=vlan_ID ] ]
SET PRIVNET NAME1=priv_name_1 IP1=priv_ip1 NAME2=priv_name_2 IP2=priv_ip2
[ SET INTERCONNECT NAME1=priv_name_1 IP1=priv_ip1 NAME2=priv_name_2
IP2=priv_ip2 ]
[ SET VCPU COUNT=guest_cpu ]
[ SET VMEM SIZE=guest_memory ]
[ SET VDISK SIZE=guest_disk ]
```

Arguments

- **NAME:** Specifies the name of the new guest VM.

Additional settings in conjunction with the `CREATE GUEST` command:

- **PARENT:** Specifies the name of the virtual machine (VM) host (Oracle Linux KVM or Oracle VM) that hosts the new guest.
- **ADMINNET :** Specifies the DNS name to use for the administration network. Also, optionally specifies the IP address, gateway, netmask, and VLAN ID for the administration network.
- **CLIENTNET :** Specifies the DNS name to use for the client network. Also, optionally specifies the IP address, gateway, netmask, and VLAN ID for the client network. May also optionally specify the host name, IP address, domain name, and netmask details to support Network Address Translation (NAT).
- **BACKUPNET :** Specifies the DNS name to use for the backup network. Also, optionally specifies the IP address, gateway, netmask, and VLAN ID for the backup network.
- **PRIVNET :** Specifies the names and IP addresses to use for the first and second private networks.
- **INTERCONNECT:** Only required if the compute nodes are configured with Exadata Secure RDMA Fabric Isolation or InfiniBand security (PKEYS). Specifies the names and IP addresses to use for the first and second private networks.
- **VCPU :** Specifies the number of virtual CPU cores allocated to the new guest VM.

- **VMEM** : Specifies the amount of RAM (in GB) allocated to the new guest VM.
- **VDISK** : Specifies the amount of disk space (in GB) allocated to the new guest VM.

Usage Notes

- For configurations with Exadata Secure RDMA Fabric Isolation or InfiniBand security (PKEYS), **INTERCONNECT** defines the private network that connects the processing nodes and **PRIVNET** defines the private storage network.
- You should save the modified XML file after the actions are merged so that you have a new XML file that reflects the addition of the new guest domain.

Example 4-26 Creating a Guest VM

This example shows how to create a new guest VM. In this example, the new guest is `exa01adm03vm04.example.com`. A series of **SET** commands are used to specify the configuration for the new guest VM.

```
oedaccli> LOAD FILE NAME=exa01adm03-pre-createing-vm04.xml
oedaccli> CREATE GUEST NAME='exa01adm03vm04.example.com'
oedaccli> SET PARENT NAME='exa01adm03.example.com'
oedaccli> SET ADMINNET NAME='exa01adm03vm04.example.com' IP='10.xxx.xx.x'
oedaccli> SET CLIENTNET NAME='exa01client03vm04.example.com' IP='10.xxx.xx.x'
oedaccli> SET PRIVNET NAME1='exa01adm03vm04-priv' IP1='192.168.16.8'
NAME2='exa01adm01vm03-priv2' IP2='192.168.16.9'
oedaccli> SET VCPU COUNT=4
oedaccli> SET VMEM SIZE=16GB
oedaccli> SET VDISK SIZE=50GB
oedaccli> SAVE ACTION
oedaccli> MERGE ACTIONS
oedaccli> SAVE FILE NAME='exa01adm03-after-createing-vm04.xml'
oedaccli> DEPLOY ACTIONS
```

Example 4-27 Creating Multiple Guests at the Same Time

This example creates the new guests `guest04` and `guest05` on the host named `host01`. Both new guests are deployed at the same time in the final **DEPLOY ACTIONS**.

```
CREATE GUEST NAME='guest04'
SET PARENT name='host01'
...
SAVE ACTION FORCE

CLONE GUEST NAME='guest05'
SET PARENT name='host01'
...
SAVE ACTION FORCE

MERGE ACTIONS
DEPLOY ACTIONS
```

4.3.18.3 DELETE GUEST

This command deletes an existing Oracle VM guest, and removes all of its configuration: instances, software installations, storage, and users from the cluster.

Syntax

```
DELETE GUEST
WHERE
  { SRCNAME=node_name [ STEPNAME=stepname ] |
    SRCNAMES='node_names' [ STEPNAME=stepname ] }
```

Arguments

- **SRCNAME** specifies the host name of an existing DomU that you want to delete.
- **SRCNAMES** specifies a comma-separated list of host names for existing guests that you want to delete. The specified list of host names must be surrounded by quotation marks.
- **STEPNAME** allows you to undo individual steps of the clone operation.

The value of *stepname* can be one of the following:

```
— ADD_INSTANCE
— EXTEND_DBHOME
— RUN_ROOTSCRIPT
— CONFIG_CLUSTERWARE
— CELL_CONNECTIVITY
— CREATE_USERS
— CREATE_GUEST
```

When used with **DELETE GUEST**, the **STEPNAME** clauses should be used in the reverse order of **CLONE GUEST** steps, and you should not skip any steps. For example, if you want to undo the **ADD_NODE** and **ADD_INSTANCE** steps, you would use the following commands:

```
DELETE GUEST WHERE SRCNAME=node_name STEPNAME=ADD_INSTANCE
DELETE GUEST WHERE SRCNAME=node_name STEPNAME=EXTEND_DBHOME
DELETE GUEST WHERE SRCNAME=node_name STEPNAME=ADD_NODE
```

Example 4-28 Deleting a Guest

The following example shows an OEDACLI session that uses the **DELETE GUEST** command along with typical output at each step.

```
oedacli> delete guest where srcname='dbm01vm103.example.com'
oedacli> save action
oedacli> merge actions
processMerge
processMergeActions
Merging Action : delete guest where srcname='dbm01vm103.example.com'
Merging DELETE GUEST
Action Validated and Merged OK
oedacli> deploy actions
Deploying Action ID : 1 delete guest where srcname='dbm01vm103.example.com'
Deploying DELETE GUEST
Delete Guest
Removing Cell Connectivity...
Deleting cellip.ora and cellinit.ora on node dbm01vm103.example.com
```

```

Done deleting cellip.ora and cellinit.ora on dbm01vm103.example.com
Successfully completed removing cell connectivity [elapsed Time [Elapsed =
5018 mS [0.0 minutes] Wed May 05 10:29:51 PDT 2021]]
Deleting users on dbm01vm103.example.com ...
Deleting cluster users on node dbm01vm103.example.com
Deleting cluster users...
Deleting groups...
Done deleting users and groups on node dbm01vm103.example.com
Successfully completed deleting users on node(s)
dbm01vm103.example.com[elapsed Time [Elapsed = 6043 mS [0.0 minutes] Wed May
05 10:29:57 PDT 2021]]
Deleting KVM Guest dbm01vm103.example.com from dom0 dbm01adm03.example.com
KVM Guest dbm01vm103.example.com deleted successfully.
Done...
Done

```

4.3.19 ILOM

You can alter or list the ILOMS in the loaded `es.xml` file.

- [ALTER ILOM](#)
- [LIST ILOMS](#)

4.3.19.1 ALTER ILOM

This command alters attributes of an ILOM.

Syntax

```

ALTER ILOM
{ DNSSERVERS='dns_servers' |
  NTPSERVERS='ntp_servers' |
  TIMEZONE=timezone } ...
WHERE
{ ID=ilom_id |
  HOSTNAME=host_name |
  ILOMNAME=ilom_host_name }

```

Arguments

You can specify the following attributes for the ILOM:

- **DNSSERVERS** : Specifies a comma-separated list of DNS servers that is enclosed in single quotes.
- **NTPSERVERS** : Specifies a comma-separated list of NTP servers that is enclosed in single quotes.
- **TIMEZONE** : Specifies a valid time zone for the ILOM.

The following arguments are available in the `WHERE` clause:

- **ID** : Specifies the `es.xml` ID for the target ILOM
- **HOSTNAME** : Specifies the current host name of the target machine
- **ILOMNAME** : Specifies the current host name of the ILOM

4.3.19.2 LIST ILOMS

This command lists the ILOMs in the `es.xml` file.

Syntax

```
LIST ILOMS
[ WHERE
    { ID = ilom_id |
      HOSTNAME = hostname |
      ILOMNAME = ilom_hostname } ]
```

Arguments

The following arguments are available in the `WHERE` clause:

- `ID` : Specifies the `es.xml` ID for the target ILOM
- `HOSTNAME` : Specifies the current host name of the target machine
- `ILOMNAME` : Specifies the current host name of the ILOM

Usage Notes

If you do not specify a `WHERE` clause, then this command lists all the ILOMs in the `es.xml` file.

4.3.20 LOGIN

You can check remote login to a machine or cluster.

- [CHECK LOGIN](#)

4.3.20.1 CHECK LOGIN

This command checks remote login to a machine or cluster.

Syntax

```
CHECK LOGIN
WHERE
    { CLUSTERNUMBER=cluster_number |
      CLUSTERNAME=cluster_name |
      CLUSTERID=cluster_id |
      HOSTNAME=host_name }
```

Arguments

The following arguments are available in the `WHERE` clause:

- `CLUSTERNUMBER` : Specifies the cluster number in the `es.xml`, starting at 1
- `CLUSTERNAME` : Specifies the name of the cluster
- `CLUSTERID` : Specifies the `es.xml` ID of the cluster
- `HOSTNAME` : Specifies the name of an individual host

4.3.21 MACHINE

You can alter or list the machines in the `es.xml` file.

- [ALTER MACHINE](#)
- [ALTER MACHINES](#)
- [LIST MACHINES](#)

4.3.21.1 ALTER MACHINE

This command modifies attributes of an Engineered System machine, which is a physical server or virtual machine (VM) guest.

Syntax

```
ALTER MACHINE
  { { DNSSERVERS='dnsservers' |
    GATEWAYADAPTER=gatewayadapter |
    HOSTNAMEADAPTER=hostnameadapter |
    NTPSERVERS='ntpservers' |
    TIMEZONE=timezone |
    VSWITCH={ true | false } |
    TYPE=newmachinetype |
    ORADISKSIZE=oradisksize |
    ORADISKPATH=oradiskpath }...
  | ACTION={ START [ WAIT={ true | false } ] |
    STOP [ WAIT={ true | false } ] |
    RESTART [ WAIT={ true | false } ] |
    CREATEIMAGE IMAGEFILE=imagefile SIZE=size
  [ FILESYSTEM=filesystem ] [ ZIPFILE=zipfile ] |
    ATTACHDISK IMAGEFILE=imagefile MOUNTPATH=mountpath
  [ FILESYSTEM=filesystem ] |
    DETACHDISK IMAGEFILE=imagefile [ DELETE=delete ] |
    SETVCPU [ VCPU=vcpu ] [ MAXVCPU=maxvcpu ] [ REBOOT={ true |
false } ] |
    SETVMEM VMEM=vmem [ REBOOT={ true | false } ] } }
WHERE
  { ID=id |
    HOSTNAME=hostname |
    CLUSTERNUMBER=clusternumber { COMPUTENUMBER=computenumber |
STORAGENUMBER=storagenumber } |
    CLUSTERNAME=clustername { COMPUTENUMBER=computenumber |
STORAGENUMBER=storagenumber } |
    CLUSTERID=clusterid { COMPUTENUMBER=computenumber |
STORAGENUMBER=storagenumber } }
```

Arguments

You can specify the following arguments for the machine:

- **DNSSERVERS:** Comma-separated list of DNS servers for the machine. Enclose the list in single quotes (').
- **GATEWAYADAPTER:** Network adapter to use for the machine gateway: ADMIN or CLIENT

- **HOSTNAMEADAPTER:** Network adapter to use for the machine hostname: `ADMIN` or `CLIENT`
- **NTPSERVERS:** Comma-separated list of NTP servers for the machine. Enclose the list in single quotes (`'`).
- **TIMEZONE:** Timezone specification for the machine
- **VSWITCH:** Specifies if the machine is connected to a vswitch. Default is `false`.
- **TYPE:** Sets a machine as new type. For example: `X8XT`
- **ORADISKSIZE:** Size (in GB) of the Oracle software volume (typically `/u01`). Default is 20, minimum is 10.
- **ORADISKPATH:** Mount point of the Oracle software volume. Default is `/u01`. For non-default values, take great care to ensure other paths are compatible with the setting.
- **ACTION:** Action to perform against the specified VM guest machine. One of the following:
 - **START:** Starts the specified guest. The action completes when the machine is ready for SSH connections. There is no error if the machine is already started.
 - **STOP:** Stops the specified guest. The action completes when the machine is shut down. There is no error if the machine is already stopped.
 - **RESTART:** Stops and then starts the specified guest. The action completes when the machine is ready for SSH connections. There is no error if the machine is already stopped.
 - **CREATEIMAGE:** Creates a disk image in the hypervisor for subsequent `ATTACHDISK` commands.
 - **ATTACHDISK:** Attaches a disk image from the hypervisor to a running guest and mounts the file system in the guest. `FILESYSTEM` must be specified if the default (`xfs`) is not used.
 - **DETACHDISK:** Unmount the file system and detach a disk image from a guest, and optionally delete the disk image.
 - **SETVCPU:** Changes the CPU resource allocation for a guest. The action allows allocation increases and reductions within the limit specified by the `maxcpu` value in the `vm.cfg` file. The action can operate on a running guest. A reboot is not required to enable changes.
 - **SETVMEM:** Changes the RAM resource allocation for a guest. A reboot is required to enable a change.

The following **ACTION** arguments are available:

- **WAIT:** Optionally specifies whether to wait for the associated action to complete. Default is `true`. With `WAIT=false`, the action starts asynchronously, and the command returns immediately with no indication or guarantee of success.
- **IMAGEFILE:** Name of the disk image file
- **SIZE:** Size (in GB) of the disk image file
- **FILESYSTEM:** File system type used in the disk image. Default is `xfs`. Alternative is `ext4`.
- **ZIPFILE:** Zip file used to optionally populate a new disk image
- **MOUNTPATH:** Guest directory used to mount the disk image
- **DELETE:** Optionally specifies whether to delete the disk image. Default is `false`.

- **VCPU:** Number of Virtual CPUs allocated to the guest machine
- **MAXVCPU:** Maximum number of Virtual CPUs allocated to the guest machine
- **VMEM:** Amount of RAM (in GB) allocated to the guest machine
- **REBOOT:** Optionally specifies whether to automatically reboot the guest. Default is `false`.

The following arguments are available in the `WHERE` clause:

- **ID:** Specifies the identifier for the target machine in the OEDA XML file (`es.xml`)
- **HOSTNAME:** Specifies the current host name of the target machine
- **CLUSTERNUMBER:** Cluster number associated with target machine in the OEDA XML file (`es.xml`), starting at 1
- **CLUSTERNAME:** Name of the cluster associated with target machine in the OEDA XML file (`es.xml`)
- **CLUSTERID:** ID of the cluster associated with target machine in the OEDA XML file (`es.xml`)
- **COMPUTENUMBER:** The compute node number associated with target machine in the OEDA XML file (`es.xml`). The numbering of compute nodes starts at 1 for each cluster.
- **STORAGENUMBER:** The storage server number associated with target machine in the OEDA XML file (`es.xml`). The numbering of storage servers starts at 1 for each cluster.

Example 4-29 ACTION Examples

The following examples show commands that use the `ACTION` argument:

- Start a guest:

```
ALTER MACHINE ACTION=START WHERE HOSTNAME=node1hostname
```

- Stop a guest:

```
ALTER MACHINE ACTION=STOP WHERE HOSTNAME=node1hostname
```

- Restart a guest:

```
ALTER MACHINE ACTION=RESTART WHERE HOSTNAME=node1hostname
```

- Create a disk image:

```
ALTER MACHINE ACTION=CREATEIMAGE IMAGEFILE=example.img SIZE=2 WHERE
HOSTNAME=node1hostname
```

```
ALTER MACHINE ACTION=CREATEIMAGE IMAGEFILE=example.img SIZE=2
ZIPFILE=data.zip WHERE HOSTNAME=node1hostname
```

```
ALTER MACHINE ACTION=CREATEIMAGE IMAGEFILE=example.img SIZE=2
FILESYSTEM=ext4 WHERE HOSTNAME=node1hostname
```

```
ALTER MACHINE ACTION=CREATEIMAGE IMAGEFILE=example.img SIZE=2
FILESYSTEM=ext4 ZIPFILE=data.zip WHERE HOSTNAME=node1hostname
```

- Attach a disk image:

```
ALTER MACHINE ACTION=ATTACHDISK IMAGEFILE=/EXAVMIMAGES/example.img
MOUNTPATH=/u02 WHERE HOSTNAME=node1hostname
```

```
ALTER MACHINE ACTION=ATTACHDISK IMAGEFILE=/EXAVMIMAGES/example.img
MOUNTPATH=/u02 FILESYSTEM=ext4 WHERE HOSTNAME=node1hostname
```

- Detach a disk image:

```
ALTER MACHINE ACTION=DETACHDISK IMAGEFILE=example.img WHERE
HOSTNAME=node1hostname
```

```
ALTER MACHINE ACTION=DETACHDISK IMAGEFILE=example.img DELETE=true WHERE
HOSTNAME=node1hostname
```

- Change the CPU resource allocation:

```
ALTER MACHINE ACTION=SETVCPU MAXVCPU=12 VCPU=12 WHERE
HOSTNAME=node1hostname
```

- Change the RAM resource allocation:

```
ALTER MACHINE ACTION=SETVMEM VMEM=9 WHERE HOSTNAME=node1hostname
```

4.3.21.2 ALTER MACHINES

This command modifies attributes of a collection of Engineered System machines.

Syntax

```
ALTER MACHINES
{ ORADISKSIZES=oradisksize |
  ORADISKPATH=oradiskpath |
  IMAGEVERSION=imageversion }...
```

```
WHERE
  { CLUSTERNUMBER=clusternumber |
    CLUSTERNAME=clustername |
    CLUSTERID=clusterid |
    TYPE=GUESTS }
```

Arguments

You can specify the following arguments for the machine:

- **ORADISKSIZE:** Size (in GB) of the Oracle software volume (typically /u01). Default is 20, minimum is 10.
- **ORADISKPATH:** Mount point of the Oracle software volume. Default is /u01. For non-default values, take great care to ensure other paths are compatible with the setting.
- **IMAGEVERSION:** The version of the image to deploy on guest machines. This argument is only applicable to guest machines (WHERE TYPE=GUEST).

The following arguments are available in the WHERE clause:

- **CLUSTERNUMBER:** Cluster number associated with target machine in the OEDA XML file (es.xml), starting at 1
- **CLUSTERNAME:** Name of the cluster associated with target machine in the OEDA XML file (es.xml)
- **CLUSTERID:** ID of the cluster associated with target machine in the OEDA XML file (es.xml)
- **TYPE=GUEST:** Applies the change to all guest machines. This option is only applicable when setting the IMAGEVERSION.

4.3.21.3 LIST MACHINES

This command lists the machine available in the es.xml file for an engineered system.

Syntax

```
LIST MACHINES
[ WHERE
  { TYPE=type |
    ID=machine_id |
    CLUSTERNUMBER=cluster_number [ TYPE=type ] |
    CLUSTERNAME=cluster_name [ TYPE=type ] |
    CLUSTERID=cluster_id [ TYPE=type ] } ]
```

Arguments

The following arguments are available in the WHERE clause:

- **TYPE :** Specifies the machine type. The valid values are COMPUTE, CELL, STORAGE, DOM0, GUEST, DOMU, KVM, or KVMGUEST.
- **ID :** Specifies the es.xml ID for an individual machine
- **CLUSTERNUMBER :** Specifies the cluster number in the es.xml, starting at 1
- **CLUSTERNAME :** Specifies the name of the cluster
- **CLUSTERID :** Specifies the es.xml ID of the cluster

Usage Notes

If you do not specify a `WHERE` clause, then this command lists all the machines defined in the `es.xml` file.

4.3.22 NETWORK

You can add, alter, delete, or list the networks defined in the XML configuration file.

- [ADD NETWORK](#)
- [ALTER NETWORK](#)
- [ALTER NETWORKS](#)
- [DELETE NETWORK](#)
- [LIST NETWORKS](#)

4.3.22.1 ADD NETWORK

This command adds a network record to a machine in the existing configuration.

Syntax

```
ADD NETWORK
  NETWORKTYPE=network_type
  HOSTNAME=hostname
  IP=ip_addr
  NETMASK=netmask
  DOMAINNAME=domain_name
  MASTER=master
  [ MTU=mtu ]
  [ GATEWAY=gateway ]
  [ SHENABLED={ true | false } ]
  [ MAC=mac_addr ]
  [ LINKSPEED=nic_speed ]
  [ SLAVE='slave' ]
  [ PKEY=pkey ]
  [ PKEYNAME=pkey_name ]
  [ STATUS = status ]
  [ LACP={ true | false } ]
  [ VLANID=vlan_id ]
  [ INTERFACENAME=interface_name ]
  [ DESCRIPTION='network_description' ]
  [ VSWITCHNETWORKPARAMS='vswitchnetwork_parameters' ]
  [ NATHOSTNAME=nat_host_name ]
  [ NATIP=nat_ip_addr ]
  [ NATDOMAINNAME=nat_domain_name ]
  [ NATNETMASK=nat_mask ]
WHERE
  { HOSTNAME=hostname |
    CLUSTERNAME=cluster_name { COMPUTENUMBER=compute_number |
      STORAGENUMBER=storage_number } |
    CLUSTERNUMBER=cluster_name { COMPUTENUMBER=compute_number |
      STORAGENUMBER=storage_number } }
```

Arguments

You can specify the following attributes when adding a network:

- **NETWORKTYPE:** Specifies the type of network, which can be `ADMIN`, `CLIENT`, `BACKUP`, `PRIVATE`, `OTHER`, `INGESTVIP`, `REPLICATIONVIP`, or `ILOM`.
- **HOSTNAME:** Specifies the short host name for the network
- **IP:** Specifies the IP address of the network
- **NETMASK:** Specifies the network netmask
- **DOMAINNAME:** Specifies the domain name of the network
- **MASTER:** Specifies the primary adapter name
- **MTU:** Specifies the network MTU (maximum transmission unit). Not applicable for `PRIVATE` or `ILOM` networks.
- **GATEWAY:** Specifies the network subnet gateway
- **SSHENABLED:** `TRUE` indicates that SSH should be enabled for this adapter
- **MAC:** Specifies the MAC address
- **LINKSPEED:** Either `10000` or `25000`. Only for X7-2 or later 2-socket compute nodes.
- **SLAVE:** Specifies a space-separated list of secondary devices for a bonded network, enclosed in single quotes
- **PKEY:** Specifies the pkey, for `PRIVATE` networks only
- **PKEYNAME:** Specifies the pkey name, for `PRIVATE` networks only
- **STATUS:** Specifies the status of the network
- **LACP:** `TRUE` indicates that LACP should be enabled
- **VLANID:** Specifies the VLAN identifier. Applicable if the network is a non-private VLAN network, or a VLAN network on a KVM guest.
- **INTERFACENAME:** The name for the private network on the VLAN for a KVM guest and cell, `clre[0-1]` or `stre[0-1]`.
- **DESCRIPTION:** The text description for this network, surrounded by single quotes.
- **VSWITCHNETWORKPARAMS:** Parameters required for vswitches, surrounded by single quotes.
- **NATHOSTNAME:** Specifies the NAT host name
- **NATIP:** Specifies the NAT IP address
- **NATDOMAINNAME:** Specifies the NAT domain name
- **NATNETMASK:** Specifies the NAT netmask

The following arguments are available in the `WHERE` clause:

- **HOSTNAME:** Specifies the host name for an existing compute node
- **CLUSTERNUMBER:** Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME:** Specifies the name of an existing cluster
- **COMPUTENUMBER:** Specifies the number of an existing compute node in the cluster, starting at 1

- **STORAGENUMBER:** Specifies the umber of an existing storage cell in the cluster, starting at 1

4.3.22.2 ALTER NETWORK

This command modifies a network record for a machine in the existing configuration.

Syntax

```
ALTER NETWORK
  { HOSTNAME=hostname |
    IP=ip_addr |
    NETMASK=netmask |
    DOMAINNAME=domain_name |
    MASTER=master |
    GATEWAY=gateway |
    SSHENABLED={ true | false } |
    MAC=mac_addr |
    MTU=mtu |
    LINKSPEED=nic_speed |
    SLAVE='slave' |
    PKEY=pkey |
    PKEYNAME=pkey_name |
    STATUS=status |
    LACP={ true | false } |
    VLANID=vlan_id |
    INTERFACENAME=interface_name |
    DESCRIPTION='network_description' |
    VSWITCHNETWORKPARAMS='vswitchnetwork_parameters' |
    NATHOSTNAME=nat_host_name |
    NATIP=nat_ip_addr |
    NATDOMAINNAME=nat_domain_name |
    NATNETMASK=nat_mask } ...
WHERE
  { ID=network_id |
    NETWORKHOSTNAME=network_hostname |
    NATHOSTNAME=nat_hostname |
    HOSTNAME=hostname NETWORKTYPE=network_type [ PRIVATEID=private_id ] |
    CLUSTERNAME=cluster_name { COMPUTENUMBER=compute_number |
    STORAGENUMBER=storage_number } NETWORKTYPE=network_type
  [ PRIVATEID=private_id ] |
    CLUSTERNUMBER=cluster_number { COMPUTENUMBER=compute_number |
    STORAGENUMBER=storage_number } NETWORKTYPE=network_type
  [ PRIVATEID=private_id ] }
```

Arguments

You can specify the following attributes when altering a network:

- **HOSTNAME:** Specifies the short host name for the network
- **IP:** Specifies the IP address of the network
- **NETMASK:** Specifies the netmask of the network
- **DOMAINNAME:** Specifies the domain name of the network
- **MASTER:** Specifies the primary adapter name

- **GATEWAY:** Specifies the network subnet gateway
- **SSHENABLED:** TRUE indicates that SSH should be enabled for this adapter
- **MAC:** Specifies the MAC address
- **MTU:** Specifies the network MTU (maximum transmission unit). Not applicable for `PRIVATE` or `ILOM` networks.
- **LINKSPEED:** Either 10000 or 25000. Only for X7-2 or later 2-socket compute nodes.
- **SLAVE:** Specifies a space-separated list of secondary devices for a bonded network, enclosed in single quotes
- **PKEY:** Specifies the pkey, for `PRIVATE` networks only
- **PKEYNAME:** Specifies the pkey name, for `PRIVATE` networks only
- **STATUS:** Specifies the status of the network
- **LACP:** TRUE indicates that LACP should be enabled
- **VLANID:** Specifies the VLAN identifier. Applicable if the network is a non-private VLAN network, or a VLAN network on a KVM guest.
- **INTERFACENAME:** The name for the private network on the VLAN for a KVM guest and cell, `clre[0-1]` or `stre[0-1]`.
- **DESCRIPTION:** The text description for this network, surrounded by single quotes.
- **VSWITCHNETWORKPARAMS:** Parameters required for vswitches, surrounded by single quotes.
- **NATHOSTNAME:** Specifies the NAT host name
- **NATIP:** Specifies the NAT IP address
- **NATDOMAINNAME:** Specifies the NAT domain name
- **NATNETMASK:** Specifies the NAT netmask

The following arguments are available in the `WHERE` clause:

- **ID:** Specifies the ID for the target network in the XML configuration file.
- **NETWORKHOSTNAME:** Specifies the host name for an network object, not the machine host name
- **NATHOSTNAME:** Specifies the NAT host name for an existing compute node
- **HOSTNAME:** Specifies the host name for an existing compute node
- **NETWORKTYPE:** Specifies the type of network, which can be `ADMIN`, `CLIENT`, `BACKUP`, `PRIVATE`, `OTHER`, `INGESTVIP`, `REPLICATIONVIP`, or `ILOM`.
- **PRIVATEID:** Specifies the ID of the private network
- **CLUSTERNUMBER:** Specifies the cluster number in the XML configuration file, starting at 1
- **CLUSTERNAME:** Specifies the name of an existing cluster
- **COMPUTENUMBER:** Specifies the number of an existing compute node in the cluster, starting at 1
- **STORAGENUMBER:** Specifies the number of an existing storage cell in the cluster, starting at 1

4.3.22.3 ALTER NETWORKS

This command modifies the VLAN number for the specified machines.

Syntax

```
ALTER NETWORKS
  VLANID=vlan_id
WHERE
  HOSTNAMES=hostnames
  NETWORKTYPE=network_type
```

Arguments

- **VLANID:** Specifies the VLAN identifier.
- **HOSTNAMES:** Specifies a comma-separated list of host names
- **NETWORKTYPE:** Specifies the type of network, which can be ADMIN, CLIENT, BACKUP, PRIVATE (storage private network), or CLUSTERPRIVATE.

Usage Notes

- During corresponding MERGE ACTIONS processing, the VLAN identifier is set even if the specified host does not have an existing VLAN.
- During corresponding DEPLOY ACTIONS processing, the storage cells reboot in parallel to enable the new VLAN identifier.

4.3.22.4 DELETE NETWORK

This command removes a network record for a machine.

Syntax

```
DELETE NETWORK
WHERE
  { ID=network_id |
    HOSTNAME=network_hostname [ NETWORKTYPE=network_type ] |
    CLUSTERNAME=cluster_name { COMPUTENUMBER=compute_number |
    STORAGENUMBER=storage_number } [ NETWORKTYPE=network_type ] |
    CLUSTERNUMBER=cluster_number { COMPUTENUMBER=compute_number |
    STORAGENUMBER=storage_number } [ NETWORKTYPE=network_type ] |
    CLUSTERID=cluster_id { COMPUTENUMBER=compute_number |
    STORAGENUMBER=storage_number } [ NETWORKTYPE=network_type ] }
```

Arguments

The following arguments are available in the WHERE clause:

- **ID :** Specifies the ID for the network object in the XML configuration file.
- **HOSTNAME :** Specifies the host name for an existing compute node
- **NETWORKTYPE :** Specifies the type of network, which can be ADMIN, CLIENT, BACKUP, PRIVATE, OTHER, INGESTVIP, REPLICATIONVIP, or ILOM.
- **CLUSTERNAME :** Specifies the name of an existing cluster
- **CLUSTERNUMBER :** Specifies the cluster number in the XML configuration file, starting at 1
- **CLUSTERID :** Specifies the cluster ID in the XML configuration file

- **COMPUTENUMBER** : Specifies the number of an existing compute node in the cluster, starting at 1
- **STORAGENUMBER** : Specifies the number of an existing storage cell in the cluster, starting at 1

4.3.22.5 LIST NETWORKS

This command lists the network details.

Syntax

```
LIST NETWORKS
[ WHERE
    { ID=network_id |
      HOSTNAMES=ALL [ NETWORKTYPE=network_type ] |
      HOSTID=host_id [ NETWORKTYPE=network_type ] |
      HOSTNAME=network_hostname [ NETWORKTYPE=network_type ] |
      CLUSTERNAME=cluster_name { COMPUTENUMBER=compute_number |
      STORAGENUMBER=storage_number } [ NETWORKTYPE=network_type ] |
      CLUSTERNUMBER=cluster_number { COMPUTENUMBER=compute_number |
      STORAGENUMBER=storage_number } [ NETWORKTYPE=network_type ] } ]
```

Arguments

The following arguments are available in the **WHERE** clause:

- **ID** : Specifies the ID for the network object in the XML configuration file
- **HOSTNAMES=ALL** : Specifies all hosts
- **NETWORKTYPE** : Specifies the type of network, which can be ADMIN, CLIENT, BACKUP, PRIVATE, or ILOM.
- **HOSTID** : Specifies the host ID in the XML configuration file
- **HOSTNAME** : Specifies the host name for an existing compute node
- **CLUSTERNAME** : Specifies the name of an existing cluster
- **CLUSTERNUMBER** : Specifies the cluster number in the XML configuration file, starting at 1
- **CLUSTERID** : Specifies the cluster ID in the XML configuration file
- **COMPUTENUMBER** : Specifies the number of an existing compute node in the cluster, starting at 1
- **STORAGENUMBER** : Specifies the number of an existing storage cell in the cluster, starting at 1

4.3.23 PASSWORD

You can specify the default passwords.

- **ALTER PASSWORD**
Modifies the OEDA properties that specifies the default passwords.

4.3.23.1 ALTER PASSWORD

Modifies the OEDA properties that specifies the default passwords.

Syntax

```
ALTER PASSWORD
  [ PASSWORD=new_password ]
WHERE
  USER={ ROOT | NONROOT }
```

Arguments

- **PASSWORD:** Specifies the new password.
- **USER:** Specifies the user type; either `ROOT` or `NONROOT`.

Usage Notes

If the `PASSWORD` argument is not specified, then the user is prompted to specify and verify the password interactively.

4.3.24 PROPERTY

You can set OEDA properties.

- **ALTER PROPERTY**
Modifies an OEDA properties parameter.
- **LIST PROPERTIES**
Displays information about OEDA properties.

4.3.24.1 ALTER PROPERTY

Modifies an OEDA properties parameter.

Syntax

```
ALTER PROPERTY
  NAME=property_name
  VALUE=value
WHERE
  [ OS=os_name ]
```

Arguments

- **NAME:** Specifies the name of the property to alter. See `LIST PROPERTIES` to display the available properties.
- **VALUE:** Specifies the new property value
- **OS:** Optionally specifies the operating system that is associated with the property. Permitted values are: `LINUX`, `SPARC`, `WINDOWS`, `MAC`, `OVM`, or `WEB`.

Usage Notes

If the `OS` argument is not specified, then the property is updated in the main `es.properties` file is updated.

4.3.24.2 LIST PROPERTIES

Displays information about OEDA properties.

Syntax

```
LIST PROPERTIES
[ WHERE
  OS=os_name ]
```

Arguments

- **OS:** Optionally specifies the operating system that is associated with the properties. Permitted values are: LINUX, SPARC, WINDOWS, MAC, OVM, or WEB.

Usage Notes

If the **OS** argument is not specified, then the properties in the main `es.properties` are displayed.

4.3.25 RACK

You can manage an Engineered System rack.

- [ADD RACK](#)
Add a new rack to the Engineered System XML file.
- [ALTER RACK](#)
Modify a rack in the Engineered System XML file.
- [LIST RACKS](#)
Displays information about racks in the Engineered System XML file.
- [LIST RACKDESCRIPTION](#)
Displays information about rack types supported by OEDA.

4.3.25.1 ADD RACK

Add a new rack to the Engineered System XML file.

Syntax

```
ADD RACK
  TYPE=ELASTIC
  MODEL={ X7 | X8 | X8M }
  SRCRACK=srcrack

SET ADMIN NAME=admin_name IP=admin_ip
SET LEAF1 NAME=leaf1_name IP=leaf1_ip
SET LEAF2 NAME=leaf2_name IP=leaf2_ip
[ SET SPINE NAME=spine_name IP=spine_ip ]
[ SET PDU1 NAME=pdu1_name IP=pdu1_ip ]
[ SET PDU2 NAME=pdu2_name IP=pdu2_ip ]
```

Arguments

- **TYPE:** Specifies the `ELASTIC` rack type.
- **MODEL:** Specifies the rack model. Permitted values are: `X7`, `X8`, or `X8M`.
- **SRCRACK:** Specifies the `es.xml` identifier of the existing rack, which is used as the template for the new rack.

You can set the following options for the new rack:

- **ADMIN :** Specifies the DNS name and IP address to use for the administration network switch. The DNS name may optionally include the domain name.
- **LEAF1 :** Specifies the DNS name and IP address to use for the first leaf switch. The DNS name may optionally include the domain name.
- **LEAF2 :** Specifies the DNS name and IP address to use for the second leaf switch. The DNS name may optionally include the domain name.
- **SPINE :** Optionally specifies the DNS name and IP address to use for the spine switch. The DNS name may optionally include the domain name.
- **PDU1 :** Optionally specifies the DNS name and IP address to use for the first PDU. The DNS name may optionally include the domain name.
- **PDU2 :** Optionally specifies the DNS name and IP address to use for the second PDU. The DNS name may optionally include the domain name.

4.3.25.2 ALTER RACK

Modify a rack in the Engineered System XML file.

Syntax

```
ALTER RACK
  ULOC={ ulocation | 0 | GENERATE }
WHERE
  RACKNUMBER=rack_number
  [ HOSTNAME=host_name ]
```

```
ALTER RACK
  { DESCRIPTION=description |
    RACKID=rack_type } ...
WHERE
  RACKNUMBER=rack_number
```

Arguments

You can modify the following attributes:

- **ULOC:** Sets the rack location (`uloc`) as follows:
 - **ULOC=*ulocation*:** Specifies a rack location for a specific host or switch
 - **ULOC=0:** Sets the rack location to 0 (zero) for all hosts and switches in the specified rack.

- `ULOC=GENERATE`: Assigns valid rack locations to all hosts and switches in the specified rack.
- `DESCRIPTION`: Sets the rack description
- `RACKID`: Sets the rack type

The following arguments are available in the `WHERE` clause:

- `RACKNUMBER`: Specifies the number of the rack in the Engineered System XML file, starting at 1
- `HOSTNAME`: Specifies the host name for a specific host or switch

Usage Notes

If you use the `ULOC` argument and supply a specific rack location (`ULOC=ulocation`), then you must specify `HOSTNAME=host_name` in the `WHERE` clause.

4.3.25.3 LIST RACKS

Displays information about racks in the Engineered System XML file.

Syntax

```
LIST RACKS
```

4.3.25.4 LIST RACKDESCRIPTION

Displays information about rack types supported by OEDA.

Syntax

```
LIST RACKDESCRIPTION
[ WHERE
  TYPE=rack ]
```

Arguments

`TYPE`: Optionally specifies the rack model that you want to use to limit the command output.

Usage Notes

If the `WHERE` clause is omitted, all rack types are displayed.

4.3.26 REQUIREDFILES

You can display information about required files.

- [LIST REQUIREDFILES](#)
Displays information about required installation files for Oracle Grid Infrastructure or Oracle Database.

4.3.26.1 LIST REQUIREDFILES

Displays information about required installation files for Oracle Grid Infrastructure or Oracle Database.

Syntax

```
LIST REQUIREDFILES
{ GRID |
  DATABASE |
  GIVERSION='<giversion>' |
  DBVERSION='<dbversion>' }
```

Arguments

- **GRID:** Displays information about required files for Oracle Grid Infrastructure
- **DATABASE:** Displays information about required files for Oracle Database
- **GIVERSION:** Specifies the Oracle Grid Infrastructure version including the release update (RU). For example: 12.1.0.2.170418.
- **DBVERSION:** Specifies the Oracle Database version including the release update (RU). For example: 12.1.0.2.170418.

Usage Notes

- Use **GIVERSION** or **DBVERSION** when no `es.xml` file is loaded.
- Use **LIST SOFTWAREVERSIONS { GRID | DATABASE }** to display information about software versions.

4.3.27 REMOTE

You can run a command on a remote host.

- [EXEC REMOTE](#)
Runs a command on remote hosts.

4.3.27.1 EXEC REMOTE

Runs a command on remote hosts.

Syntax

```
EXEC REMOTE
  COMMAND='command'
WHERE
  HOSTNAMES='host_names'
```

Section Title

- **COMMAND:** Specifies the command to run on the specified hosts.
- **HOSTNAMES:** Specifies the comma-separated list of remote hosts.

4.3.28 SCAN

You can add, alter, delete, or list the SCANS for a cluster.

- [ADD SCAN](#)

- [ALTER SCAN](#)
- [DELETE SCAN](#)
- [LIST SCANS](#)

4.3.28.1 ADD SCAN

This command adds a SCAN to the an Oracle Grid Infrastructure cluster.

Syntax

```
ADD SCAN
  SCANNAME=scan_name
  [ SCANPORT=scan_port ]
  SCANIPS='scan_ip_addrs'
WHERE
  { CLUSTERNAME=cluster_name |
    CLUSTERNUMBER=cluster_number |
    CLUSTERID=cluster_id }
```

Arguments

You can specify the following attributes for the new SCAN:

- **SCANNAME** : Specifies the DNS name for the SCAN
- **SCANPORT** : Optionally specifies the port for the SCAN. The default value is 1521.
- **SCANIPS** : A comma-separated list of IP addresses for the SCAN, enclosed in single quotes

The following arguments are available in the **WHERE** clause:

- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERNUMBER** : Specifies the cluster number in the XML configuration file, starting at 1
- **CLUSTERID** : Specifies the ID of the cluster in the XML configuration file

4.3.28.2 ALTER SCAN

This command modifies the attributes of a SCAN for an Oracle Grid Infrastructure cluster.

Syntax

```
ALTER SCAN
  { SCANNAME=scan_name |
    SCANPORT=scan_port |
    SCANIPS='scan_ip_addrs' } ...
WHERE
  { CLUSTERNAME=cluster_name |
    CLUSTERNUMBER=cluster_number |
    CLUSTERID=cluster_id }
```

Arguments

You can modify the following attributes for the SCAN:

- **SCANNAME** : Specifies the DNS name for the SCAN

- **SCANPORT** : Specifies the port for the SCAN
- **SCANIPS** : A comma-separated list of IP addresses for the SCAN, enclosed in single quotes

The following arguments are available in the **WHERE** clause:

- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERNUMBER** : Specifies the cluster number in the XML configuration file, starting at 1
- **CLUSTERID** : Specifies the ID of the cluster in the XML configuration file

4.3.28.3 DELETE SCAN

This command removes the SCAN for an Oracle Grid Infrastructure cluster.

Syntax

```
DELETE SCAN
WHERE
  { CLUSTERNAME=cluster_name |
    CLUSTERNUMBER=cluster_number |
    CLUSTERID=cluster_id }
```

Arguments

The following arguments are available in the **WHERE** clause:

- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERNUMBER** : Specifies the cluster number in the XML configuration file, starting at 1
- **CLUSTERID** : Specifies the ID of the cluster in the XML configuration file

4.3.28.4 LIST SCANS

This command lists the SCAN details for all Oracle Grid Infrastructure clusters or for an individual cluster.

Syntax

```
LIST SCANS
[ WHERE
  { ID=scan_id |
    CLUSTERNAME=cluster_name |
    CLUSTERNUMBER=cluster_number |
    CLUSTERID=cluster_id } ]
```

Arguments

The following arguments are available in the **WHERE** clause:

- **ID** : Specifies the ID of the SCAN object in the XML configuration file
- **CLUSTERNAME** : Specifies the name of the cluster
- **CLUSTERNUMBER** : Specifies the cluster number in the XML configuration file, starting at 1
- **CLUSTERID** : Specifies the ID of the cluster in the XML configuration file

Usage Notes

If you do not specify a WHERE clause, then this command lists details about all the SCANS for all the clusters in the XML configuration file.

4.3.29 SECURITY

You can manage security settings.

- **SET SECURITY**
This command performs security-related actions, like setting the non-default root password for the OEDACLI session or controlling password-based authentication.

4.3.29.1 SET SECURITY

This command performs security-related actions, like setting the non-default root password for the OEDACLI session or controlling password-based authentication.

Prerequisites

Before you can use this command with the `ENABLEPASSWORDLOGIN` option, you must configure SSH key-based authentication on the target system. See [DEPLOY SSHKEYS](#).

Syntax

```
SET SECURITY
  PASSWORD=root_password

SET SECURITY
  ENABLEPASSWORDLOGIN={ true | false }
  [ HOSTNAMES="host1[,host2[,host3 ...]]" ]
```

Arguments

- `PASSWORD`: Specifies the non-default root password for the OEDACLI session
- `ENABLEPASSWORDLOGIN`: Specifies whether to enable or disable password-based authentication for the `root` user
- `HOSTNAMES`: Optionally specifies the list of host names on which to act

Usage Notes

If the `HOSTNAMES` argument is not specified, then the command acts on the currently loaded Engineered System XML file.

4.3.30 SOFTWAREVERSIONS

You can check supported software versions.

- **LIST SOFTWAREVERSIONS**
Displays information about supported versions of Oracle Grid Infrastructure and Oracle Database.

4.3.30.1 LIST SOFTWAREVERSIONS

Displays information about supported versions of Oracle Grid Infrastructure and Oracle Database.

Syntax

```
LIST SOFTWAREVERSIONS [ GRID | DATABASE ]
```

Arguments

- **GRID:** Displays information about supported versions of Oracle Grid Infrastructure
- **DATABASE:** Displays information about supported versions of Oracle Database

Usage Notes

- If neither argument (GRID or DATABASE) is specified, then `LIST SOFTWAREVERSIONS` displays information about supported software versions for Oracle Grid Infrastructure and Oracle Database.

4.3.31 SSHKEYS

You can generate and deploy SSH keys for `root` user authentication on Oracle Exadata Rack.

- **DEPLOY SSHKEYS**
This command configures SSH key-based authentication on all hosts in the Engineered System XML file (`es.xml`), or the host names specified in the command.
- **GENERATE SSHKEYS**
This command generates SSH key pairs for all hosts in the Engineered System XML file (`es.xml`), or the host names specified in the command. The keys are saved in the `OEDA WorkDir`.
- **SET SSHKEYS**
This command enables and disables SSH key-based authentication for remote commands run by OEDACLI.

4.3.31.1 DEPLOY SSHKEYS

This command configures SSH key-based authentication on all hosts in the Engineered System XML file (`es.xml`), or the host names specified in the command.

Syntax

```
DEPLOY SSHKEYS
[ PASSWORD=root_password ]
[ ENABLEPASSWORDLOGIN={ true | false } ]
[ HOSTNAMES="host1[,host2[,host3 ...]]" ]
```

Arguments

- **PASSWORD:** Optionally specifies the non-default root password for connecting to the hosts to set up SSH key access

- **ENABLEPASSWORDLOGIN:** Optionally specifies whether to enable or disable password-based authentication for the `root` user after SSH key access is set up
- **HOSTNAMES:** Optionally specifies the list of host names on which to deploy SSH keys

Usage Notes

- If the SSH key pairs are not located in the OEDA `WorkDir`, then the command also generates them.
- If the `HOSTNAMES` argument is not specified, then the command acts on the currently loaded Engineered System XML file.

Example 4-30 Deploy SSH keys for all hosts of an XML file

```
oedaccli> LOAD FILE NAME=es.xml
SUCCESS - file loaded
oedaccli> DEPLOY SSHKEYS ENABLEPASSWORDLOGIN=false
```

Example 4-31 Deploy SSH keys for specified hosts

```
oedaccli> DEPLOY SSHKEYS ENABLEPASSWORDLOGIN=false PASSWORD="password"
HOSTNAMES="dbm01adm05.example.com,dbm01adm06.example.com"
```

4.3.31.2 GENERATE SSHKEYS

This command generates SSH key pairs for all hosts in the Engineered System XML file (`es.xml`), or the host names specified in the command. The keys are saved in the OEDA `WorkDir`.

Syntax

```
GENERATE SSHKEYS [ HOSTNAMES="host1[,host2[,host3 ...]]" ]
```

Arguments

- **HOSTNAMES** optionally specifies the list of host names for which to generate SSH keys.

Usage Notes

- If the `HOSTNAMES` argument is not specified, then the command acts on the currently loaded Engineered System XML file.

Example 4-32 Generate SSH keys for all hosts of an XML file

```
oedaccli> LOAD FILE NAME=es.xml
SUCCESS - file loaded
oedaccli> GENERATE SSHKEYS
Creating SSH Key Pairs for hosts [dbm01adm01.example.com,
dbm01adm02.example.com, ...]...
Generated SSH Keys for [dbm01adm01.example.com, dbm01adm02.example.com, ...]
```

Example 4-33 Generate SSH keys for specified hosts

```
oedaccli> GENERATE SSHKEYS
HOSTNAMES="dbm02adm05.example.com,dbm02adm06.example.com"
Creating SSH Key Pairs for hosts [dbm02adm06.example.com,
dbm02adm05.example.com]...
Generated SSH Keys for [dbm02adm06.example.com, dbm02adm05.example.com]
```

4.3.31.3 SET SSHKEYS

This command enables and disables SSH key-based authentication for remote commands run by OEDACLI.

Prerequisites

Before you can use this command, you must configure SSH key-based authentication on the target system. See [DEPLOY SSHKEYS](#).

Syntax

```
SET SSHKEYS ENABLE={ true | false }
```

Arguments

- **ENABLE** specifies whether or not to enable SSH key-based authentication for remote commands run by OEDACLI.

4.3.32 STEPS

You can display the steps for a virtual `CREATE CLUSTER` deployment.

- [LIST STEPS](#)
This command lists the steps for a virtual `CREATE CLUSTER` deployment.

4.3.32.1 LIST STEPS

This command lists the steps for a virtual `CREATE CLUSTER` deployment.

Each step is named, for example `Create Virtual Machine`. Each step also has a one word identifier, for example, `CREATEVM`. The short name identifier is used when specifying `STEPS` or `SKIPSTEPS` in other commands, such as the `CREATE CLUSTER` command.

The list of valid steps for a cluster depends on the features selected for the cluster. The steps listed are valid for the identified cluster. If the XML file contains only 1 cluster, then there is no need to identify the cluster explicitly.

The step numbers listed may change depending on features implemented for the cluster, but the short name identifier is always the same for a given step.

Syntax

```
LIST STEPS
[ WHERE
  { CLUSTERNUMBER=cluster_number |
```

```
CLUSTERNAME=cluster_name |  
ID=cluster_id } ]
```

Arguments

The following arguments are available in the `WHERE` clause:

- `CLUSTERNUMBER` : Specifies the cluster number in the `es.xml`, starting at 1
- `CLUSTERNAME` : Specifies the name of the cluster
- `ID` : Specifies the `es.xml` ID of the cluster

Example 4-34 Listing the Steps for a Specific Cluster

```
oedaccli> LIST STEPS WHERE CLUSTERNAME = Cluster-cl  
{  
  "step" : [ {  
    "stepName" : "Validate Configuration File",  
    "shortName" : "VALIDATE",  
    "id" : "1"  
  }, {  
    "stepName" : "Create Virtual Machine",  
    "shortName" : "CREATEVM",  
    "id" : "2"  
  }, {  
    "stepName" : "Create Users",  
    "shortName" : "USERS",  
    "id" : "3"  
  }, {  
    "stepName" : "Setup Cell Connectivity",  
    "shortName" : "CELLCONN",  
    "id" : "4"  
  }, {  
    "stepName" : "Calibrate Cells",  
    "shortName" : "CALIBRATE",  
    "id" : "5"  
  }, {  
    "stepName" : "Create Cell Disks",  
    "shortName" : "CELDDISKS",  
    "id" : "6"  
  }, {  
    "stepName" : "Create Grid Disks",  
    "shortName" : "GRIDDISKS",  
    "id" : "7"  
  }, {  
    "stepName" : "Install Cluster Software",  
    "shortName" : "INSTALLGI",  
    "id" : "8"  
  }, {  
    "stepName" : "Initialize Cluster Software",  
    "shortName" : "INITGI",  
    "id" : "9"  
  }, {  
    "stepName" : "Install Database Software",  
    "shortName" : "INSTALLDB",  
    "id" : "10"
```



```

    }, {
      "stepName" : "Relink Database with RDS",
      "shortName" : "RELINK",
      "id" : "11"
    }, {
      "stepName" : "Create ASM Diskgroups",
      "shortName" : "ASMDG",
      "id" : "12"
    }, {
      "stepName" : "Create Databases",
      "shortName" : "DATABASE",
      "id" : "13"
    }, {
      "stepName" : "Apply Security Fixes",
      "shortName" : "APPLYFIXES",
      "id" : "14"
    }, {
      "stepName" : "Install Exachk",
      "shortName" : "EXACHK",
      "id" : "15"
    }, {
      "stepName" : "Create Installation Summary",
      "shortName" : "INSTALLSUMMARY",
      "id" : "16"
    }, {
      "stepName" : "Resecure Machine",
      "shortName" : "RESECURE",
      "id" : "17"
    } ]
  }
}

```

4.3.33 STORAGEPOOL

- [ADD STORAGEPOOL](#)
This command adds a new Exascale storage pool.
- [ALTER STORAGEPOOL](#)
This command alters an existing Exascale storage pool.
- [DELETE STORAGEPOOL](#)
This command deletes an existing Exascale storage pool.
- [LIST STORAGEPOOL](#)
This command displays details for an existing Exascale storage pool.
- [LIST STORAGEPOOLS](#)
This command displays details for existing Exascale storage pools.

4.3.33.1 ADD STORAGEPOOL

This command adds a new Exascale storage pool.

Syntax

```

ADD STORAGEPOOL
NAME = storagepoolname

```

```

[ SIZE = size ]
[ TYPE = storagetype ]
CELLLIST = 'celllist'
[ WHERE
    EXASCALECLUSTER = exascaleclustername ]

```

Arguments

- **NAME:** Specifies the name of the Exascale storage pool.
- **SIZE:** Optionally, specifies the size of the storage pool in GB. If not specified, then the storage pool receives all of the available space in the specified cells.
- **TYPE:** Specify **EF** to create a flash-based storage pool using High Capacity Exadata storage servers. Otherwise, the storage pool media type is derived from the Exadata storage server type.
- **CELLLIST:** Specifies the list of Exadata storage servers to use for the storage pool. The value is a space-separated list of cell names, or use **ALL** to specify all available cells in the OEDA configuration file (`es.xml`). Note that the value is also surrounded by single quotes (as displayed in the command syntax).

The following argument is available in the **WHERE** clause:

- **EXASCALECLUSTER:** Specifies the name for the Exascale cluster that contains the new storage pool.

Usage Notes

- When you use the **ADD STORAGEPOOL** command, there is no immediate check to validate the specified storage pool size. However, deployment fails with an error if there is insufficient space.
- The **WHERE** clause is optional if the OEDA configuration file (`es.xml`) contains only one Exascale cluster. Otherwise, the **WHERE** clause is mandatory.

4.3.33.2 ALTER STORAGEPOOL

This command alters an existing Exascale storage pool.

Syntax

```

ALTER STORAGEPOOL
[ NAME = storagepoolname ]
[ SIZE = size ]
[ TYPE = storagetype ]
[ ADDCELLS = 'addcelllist' ]
[ DROPCELLS = 'dropcelllist' ]
WHERE
[ EXASCALECLUSTER = exascaleclustername ]
NAME = storagepoolname

```

Arguments

- **NAME:** Changes the name of the Exascale storage pool.
- **SIZE:** Changes the size (in GB) of the storage pool.

- **TYPE:** Specify `EF` to create a flash-based storage pool using High Capacity Exadata storage servers. Otherwise, the storage pool media type is derived from the Exadata storage server type.
- **ADDCELLS:** Specifies the list of Exadata storage servers to add to the storage pool. The value is a space-separated list of cell names. Note that the value is also surrounded by single quotes (as displayed in the command syntax).
- **DROPCELLS:** Specifies the list of Exadata storage servers to remove from the storage pool. The value is a space-separated list of cell names. Note that the value is also surrounded by single quotes (as displayed in the command syntax).

The following arguments are available in the `WHERE` clause:

- **EXASCALECLUSTER:** Specifies the name of the Exascale cluster that contains the storage pool. This argument is optional if the OEDA configuration file (`es.xml`) contains only one Exascale cluster.
- **NAME:** Specifies the name for the Exascale storage pool that you want to alter.

Usage Notes

- When you use the `ALTER STORAGEPOOL` command, there is no immediate check to validate the specified storage pool size. However, deployment fails with an error if there is insufficient space.

4.3.33.3 DELETE STORAGEPOOL

This command deletes an existing Exascale storage pool.

Syntax

```
DELETE STORAGEPOOL
WHERE
  [ EXASCALECLUSTER = exascaleclustername ]
  NAME = storagepoolname
```

Arguments

The following arguments are available in the `WHERE` clause:

- **EXASCALECLUSTER:** Specifies the name of the Exascale cluster that contains the storage pool. This argument is optional if the OEDA configuration file (`es.xml`) contains only one Exascale cluster.
- **NAME:** Specifies the name for the Exascale storage pool that you want to delete.

Usage Notes

- You cannot delete an Exascale storage pool while it provides resources to Oracle Grid Infrastructure or an Oracle database.

4.3.33.4 LIST STORAGEPOOL

This command displays details for an existing Exascale storage pool.

Syntax

```
LIST STORAGEPOOL
WHERE
  [ EXASCALECLUSTER = exascaleclustername ]
  NAME = storagepoolname
```

Arguments

The following arguments are available in the `WHERE` clause:

- **EXASCALECLUSTER:** Specifies the name of the Exascale cluster that contains the storage pool. This argument is optional if the OEDA configuration file (`es.xml`) contains only one Exascale cluster.
- **NAME:** Specifies the name for the Exascale storage pool that you want to display.

4.3.33.5 LIST STORAGEPOOLS

This command displays details for existing Exascale storage pools.

Syntax

```
LIST STORAGEPOOLS
[ WHERE
  EXASCALECLUSTER = exascaleclustername ]
```

Arguments

The following arguments are available in the `WHERE` clause:

- **EXASCALECLUSTER:** Specifies the name of an existing Exascale cluster.

4.3.34 STORAGETEMPLATES

- [LIST STORAGETEMPLATES](#)
This command displays details for the Exascale storage templates.

4.3.34.1 LIST STORAGETEMPLATES

This command displays details for the Exascale storage templates.

Syntax

```
LIST STORAGEPOOLTEMPLATES
```

4.3.35 SWITCH

You can display and manage the switches.

- [ADD SWITCH](#)
Add a new switch to a rack in the Engineered System XML file.

- [ALTER SWITCH](#)
- [LIST SWITCHES](#)

4.3.35.1 ADD SWITCH

Add a new switch to a rack in the Engineered System XML file.

Syntax

```
ADD SWITCH
WHERE
    { RACKID=rack_id |
      RACKNUMBER=rack_number }

SET ADMINNET NAME=admin_name, IP=admin_ip [ , GATEWAY=gateway,
NETMASK=netmask ]
SET DESCRIPTION VALUE=description
[ SET IBPARTITIONMEMBERSHIP VALUE=pkey_membership ]
[ SET ULOC VALUE=ulocation ]
[ SET TIMEZONE VALUE=timezone ]
[ SET DNSSERVERS IP1=dns_ip1, IP2=dns_ip2, IP3=dns_ip3 ]
[ SET NTPSERVERS IP1=ntp_ip1, IP2=ntp_ip2, IP3=ntp_ip3 ]
```

Arguments

The following arguments are available in the `WHERE` clause:

- **RACKID:** Specifies the rack identifier in the XML configuration file
- **RACKNUMBER:** Specifies the rack number in the XML configuration file

You can set the following options for the new rack:

- **ADMINNET:** Specifies the details for the administration network.
- **DESCRIPTION:** Specifies a description for the switch
- **IBPARTITIONMEMBERSHIP:** Specifies the switch pkey membership
- **ULOC:** Specifies physical rack location for the switch
- **TIMEZONE:** Specifies a valid time zone for the switch
- **DNSSERVERS:** Specifies the DNS servers to set in the switch configuration
- **NTPSERVERS:** Specifies the NTP servers to set in the switch configuration

4.3.35.2 ALTER SWITCH

This command alters attributes of an individual switch.

Syntax

```
ALTER SWITCH
    { DNSSERVERS='dns_servers' |
      NTPSERVERS='ntp_servers' |
      TIMEZONE=time_zone |
      DESCRIPTION=description |
```

```

        ULOC=ulocation |
        IBPARTITIONMEMBERSHIP=pkey_membership } ...
WHERE
    { ID=switch_id |
      HOSTNAME=switch_name }

```

Arguments

You can modify the following attributes for a switch:

- **DNSSERVERS:** Specifies a comma-separated list of DNS servers that is enclosed in single quotes
- **NTPSERVERS:** Specifies a comma-separated list of NTP servers that is enclosed in single quotes
- **TIMEZONE:** Specifies a valid time zone for the switch
- **DESCRIPTION:** Specifies a description for the switch
- **ULOC:** Specifies physical rack location for the switch
- **IBPARTITIONMEMBERSHIP:** Specifies the switch pkey membership

The following arguments are available in the `WHERE` clause:

- **ID:** Specifies the switch identifier in the XML configuration file
- **HOSTNAME:** Specifies the host name of the switch

4.3.35.3 LIST SWITCHES

This command displays details for all switches or for an individual switch.

Syntax

```

LIST SWITCHES
[ WHERE
    { ID=switch_id |
      HOSTNAME=switch_name } ]

```

Arguments

The following arguments are available in the `WHERE` clause:

- **ID :** Specifies the switch identifier in the XML configuration file
- **HOSTNAME :** Specifies the host name of the switch

Usage Notes

If you do not specify a `WHERE` clause, then the command displays the details for all switches.

4.3.36 SU

Controls use of the `su` command.

- **SET SU**
This command enables and disables using `su` from the `root` user to run remote non-`root` commands from OEDACLI.

4.3.36.1 SET SU

This command enables and disables using `su` from the `root` user to run remote non-`root` commands from OEDACLI.

Syntax

```
SET SU ENABLE={ true | false }
```

Arguments

- **ENABLE:** Specifies whether or not OEDACLI uses `su` from the `root` user to run remote non-`root` commands.

4.3.37 ULOC

You can display the location of rack items.

- [LIST ULOC](#)

4.3.37.1 LIST ULOC

This command displays the location of defined rack items and new rack items.

Syntax

```
LIST ULOC
[ NEWCOMPUTES={ new_computes_count | 'new_computes_list' } ]
[ NEWCELLS={ new_cells_count | 'new_cells_list' } ]
[ WHERE
    RACKNUMBER=rack_number ]
```

Arguments

- **NEWCOMPUTES:** Optionally specifies the number of new compute nodes to be added, or a comma-separated list of new compute node host names. Default is 0.
- **NEWCELLS:** Optionally specifies the number of new storage cells to be added, or a comma-separated list of new storage cell host names. Default is 0.

The following arguments are available in the `WHERE` clause:

- **RACKNUMBER:** Specifies the rack number in the XML configuration file

Usage Notes

If you do not specify a `WHERE` clause, then the command displays the details for rack number 1.

4.3.38 VAULT

- [ADD VAULT](#)
This command adds a new Exascale vault.

- **ALTER VAULT**
This command alters an existing Exascale vault.
- **DELETE VAULT**
This command deletes an existing Exascale vault.
- **LIST VAULT**
This command displays details for an existing Exascale vault.
- **LIST VAULTS**
This command displays details for existing Exascale vaults.

4.3.38.1 ADD VAULT

This command adds a new Exascale vault.

Syntax

```
ADD VAULT
  NAME = vaultname
  [ HC = hcsizesize ]
  [ EF = efsize ]
[ WHERE
  EXASCALECLUSTER = exascaleclustername ]
```

Arguments

- **NAME:** Specifies the name of the Exascale vault.
- **HC:** Optionally, provisions the vault with the specified amount of storage space in the High Capacity storage pool. You can specify a size in GB or TB, or specify a percentage of the High Capacity storage pool.
- **EF:** Optionally, provisions the vault with the specified amount of storage space in the Extreme Flash storage pool. You can specify a size in GB or TB, or specify a percentage of the Extreme Flash storage pool.

The following argument is available in the `WHERE` clause:

- **EXASCALECLUSTER:** Specifies the name for the Exascale cluster that contains the new vault.

Usage Notes

- If you specify any combination of the size arguments (`HC`, `EF`), then the vault is provisioned to use only the specified storage resources. If you do not specify any size arguments, then the vault is provisioned with unlimited access to all of the available storage pools.
- The `WHERE` clause is optional if the OEDA configuration file (`es.xml`) contains only one Exascale cluster. Otherwise, the `WHERE` clause is mandatory.

4.3.38.2 ALTER VAULT

This command alters an existing Exascale vault.

Syntax

```
ALTER VAULT
  [ NAME = vaultname ]
  [ HC = hcsizesize ]
```



```
[ EF = efsize ]
WHERE
  [ EXASCALECLUSTER = exascaleclustername ]
  NAME = vaultname
```

Arguments

- **NAME:** Changes the name of the Exascale vault.
- **HC:** Optionally, provisions the vault with the specified amount of storage space in the High Capacity storage pool. You can specify a size in GB or TB, or specify a percentage of the High Capacity storage pool.
- **EF:** Optionally, provisions the vault with the specified amount of storage space in the Extreme Flash storage pool. You can specify a size in GB or TB, or specify a percentage of the Extreme Flash storage pool.

The following arguments are available in the `WHERE` clause:

- **EXASCALECLUSTER:** Specifies the name of the Exascale cluster that contains the vault. This argument is optional if the OEDA configuration file (`es.xml`) contains only one Exascale cluster.
- **NAME:** Specifies the name of the Exascale vault that you want to alter.

Usage Notes

- If you alter any of the size arguments (**HC**, **EF**), then ensure that you set all of the required size arguments.

4.3.38.3 DELETE VAULT

This command deletes an existing Exascale vault.

Syntax

```
DELETE VAULT
WHERE
  [ EXASCALECLUSTER = exascaleclustername ]
  NAME = vaultname
```

Arguments

The following arguments are available in the `WHERE` clause:

- **EXASCALECLUSTER:** Specifies the name of the Exascale cluster that contains the vault. This argument is optional if the OEDA configuration file (`es.xml`) contains only one Exascale cluster.
- **NAME:** Specifies the name of the Exascale vault that you want to delete.

Usage Notes

- You cannot delete an Exascale vault while it provides resources to Oracle Grid Infrastructure or an Oracle database.

4.3.38.4 LIST VAULT

This command displays details for an existing Exascale vault.

Syntax

```
LIST VAULT
WHERE
  [ EXASCALECLUSTER = exascaleclustername ]
  NAME = vaultname
```

Arguments

The following arguments are available in the `WHERE` clause:

- **EXASCALECLUSTER:** Specifies the name of the Exascale cluster that contains the vault. This argument is optional if the OEDA configuration file (`es.xml`) contains only one Exascale cluster.
- **NAME:** Specifies the name of the Exascale vault that you want to display.

4.3.38.5 LIST VAULTS

This command displays details for existing Exascale vaults.

Syntax

```
LIST VAULTS
[ WHERE
  EXASCALECLUSTER = exascaleclustername
  [ STORAGEPOOL = storagepoolname ]]
```

Arguments

The following arguments are available in the `WHERE` clause:

- **EXASCALECLUSTER:** Specifies the name of the Exascale cluster that contains the vault.
- **STORAGEPOOL:** Specifies the name of the Exascale storage pool for which you want to display associated vaults.

4.3.39 VIP

You can add, alter, delete, or list the Oracle Clusterware VIPs.

- [ADD VIP](#)
- [ALTER VIP](#)
- [DELETE VIP](#)
- [LIST VIPS](#)

4.3.39.1 ADD VIP

This command adds a VIP for an individual node in a cluster.

Purpose

This command does not add additional VIPs. You should only use this command after previously issuing a `DELETE VIP` command.

Syntax

```
ADD VIP
  NAME=vip_name
  DOMAINNAME=domain_name
  IP=vip_ip_addr
WHERE
  { HOSTNAME=host_name |
    CLUSTERNAME=cluster_name COMPUTENUMBER=compute_number |
    CLUSTERNUMBER=cluster_number COMPUTENUMBER=compute_number |
    CLUSTERID=cluster_id COMPUTENUMBER=compute_number }
```

Arguments

You must specify the following attributes when adding a VIP:

- **NAME:** Specifies the short DNS name for the VIP
- **DOMAINNAME:** Specifies the domain name for the VIP
- **IP:** Specifies the IP address for the VIP

The following arguments are available in the `WHERE` clause:

- **HOSTNAME:** Specifies the host name for the compute node in the cluster
- **CLUSTERNUMBER:** Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **CLUSTERID:** Specifies the `es.xml` ID of the cluster
- **COMPUTENUMBER:** Specifies the compute node number in the cluster, starting at 1

Usage Notes

When adding a VIP to a compute node, the node must not already have a VIP defined.

4.3.39.2 ALTER VIP

This command alters the attributes of a VIP for an individual node in a cluster.

Syntax

```
ALTER VIP
  { NAME=vip_name |
    DOMAINNAME=domain_name |
    IP=vip_ip_addr } ...
WHERE
```

```
{ HOSTNAME=host_name |
  CLUSTERNAME=cluster_name COMPUTENUMBER=compute_number |
  CLUSTERNUMBER=cluster_number COMPUTENUMBER=compute_number |
  CLUSTERID=cluster_id COMPUTENUMBER=compute_number }
```

Arguments

You can modify the following attributes for the VIP:

- **NAME:** Specifies the short DNS name for the VIP
- **DOMAINNAME:** Specifies the domain name for the VIP
- **IP:** Specifies the IP address for the VIP

The following arguments are available in the `WHERE` clause:

- **HOSTNAME:** Specifies the host name for the compute node in the cluster
- **CLUSTERNUMBER:** Specifies the cluster number in the `es.xml`, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **CLUSTERID:** Specifies the `es.xml` ID of the cluster
- **COMPUTENUMBER:** Specifies the compute node number in the cluster, starting at 1

4.3.39.3 DELETE VIP

This command removes a VIP from an individual node in a cluster.

Syntax

```
DELETE VIP
WHERE
  { ID=vip_id |
    HOSTNAME=host_name |
    CLUSTERNAME=cluster_name COMPUTENUMBER=compute_number |
    CLUSTERNUMBER=cluster_number COMPUTENUMBER=compute_number |
    CLUSTERID=cluster_id COMPUTENUMBER=compute_number }
```

Arguments

The following arguments are available in the `WHERE` clause:

- **ID:** Specifies the ID for the VIP in the XML configuration file
- **HOSTNAME:** Specifies the host name for the compute node in the cluster
- **CLUSTERNUMBER:** Specifies the cluster number in the XML configuration file, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **CLUSTERID:** Specifies the ID of the cluster in the XML configuration file
- **COMPUTENUMBER:** Specifies the compute node number in the cluster, starting at 1

4.3.39.4 LIST VIPS

This command lists the VIPs for all clusters or for an individual node in a cluster.

Syntax

```
LIST VIPS
[ WHERE
    { HOSTNAME=host_name |
      CLUSTERNAME=cluster_name [ COMPUTENUMBER=compute_number ] |
      CLUSTERNUMBER=cluster_number [ COMPUTENUMBER=compute_number ] |
      CLUSTERID=cluster_id [ COMPUTENUMBER=compute_number ] }
```

Arguments

The following arguments are available in the `WHERE` clause:

- **ID:** Specifies the ID for the VIP in the XML configuration file
- **HOSTNAME:** Specifies the host name for the compute node in the cluster
- **CLUSTERNUMBER:** Specifies the cluster number in the XML configuration file, starting at 1
- **CLUSTERNAME:** Specifies the name of the cluster
- **CLUSTERID:** Specifies the ID of the cluster in the XML configuration file
- **COMPUTENUMBER:** Specifies the compute node number in the cluster, starting at 1

Usage Notes

If you do not specify a `WHERE` clause, then this command lists the VIPs for all clusters.

4.3.40 VOLUME

You can display and manage storage volumes.

- [ADD VOLUME](#)
- [DELETE VOLUME](#)
- [LIST VOLUMES](#)

4.3.40.1 ADD VOLUME

This command adds a storage volume to a KVM guest.

Syntax

```
ADD VOLUME
    MOUNTPATH=mount_path
    SIZE=size
WHERE
    HOSTNAME=host_name
```

Arguments

You must specify the following attributes when adding a volume:

- **MOUNTPATH:** Specifies the mount point directory for the volume
- **SIZE:** Specifies the volume size in GB. Must be an integer value.

The following arguments are available in the `WHERE` clause:

- `HOSTNAME`: Specifies the host name for the KVM guest

Usage Notes

You can use this command to add a custom sized volume for the `/u01` filesystem.

4.3.40.2 DELETE VOLUME

This command deletes a storage volume from a KVM guest.

Syntax

```
DELETE VOLUME
WHERE
  HOSTNAME=host_name
  MOUNTPATH=mount_path
```

Arguments

The following arguments are required in the `WHERE` clause:

- `HOSTNAME`: Specifies the host name for the KVM guest
- `MOUNTPATH`: Specifies the mount point directory for the volume

4.3.40.3 LIST VOLUMES

This command lists the volumes defined for an individual KVM guest.

Syntax

```
LIST VOLUMES
WHERE
  HOSTNAME=host_name
```

Arguments

The following arguments are required in the `WHERE` clause:

- `HOSTNAME`: Specifies the host name for the KVM guest

4.3.41 XMLACTION

You can manage actions relating to the Engineered System XML file.

- [ALTER XMLACTION](#)
- [DELETE XMLACTION](#)
- [LIST XMLACTION](#)
- [LIST XMLACTIONS](#)

4.3.41.1 ALTER XMLACTION

This command alters a specific OEDACLI action.

Syntax

```
ALTER XMLACTION
  ID=action_identifer
  CMDID=command_identifer
  CMD=command
```

Arguments

- ID: Specifies the action identifier. See output from LIST XMLACTIONS.
- CMDID: Specifies the command identifier. See output from LIST XMLACTIONS.
- CMD: Specifies the command for the action.

4.3.41.2 DELETE XMLACTION

This command removes a specific OEDACLI action.

Syntax

```
DELETE XMLACTION
  ID=action_identifer
```

Arguments

ID: Specifies the action identifier. See output from LIST XMLACTIONS.

4.3.41.3 LIST XMLACTION

This command lists a specific OEDACLI action.

Syntax

```
LIST XMLACTION
  ID=action_identifer
```

Arguments

ID: Specifies the action identifier. See output from LIST XMLACTIONS.

4.3.41.4 LIST XMLACTIONS

This command lists the current actions created by OEDACLI.

Syntax

```
LIST XMLACTIONS
```