Oracle Restart

Objectives

After completing this lesson, you should be able to use Oracle Restart to manage components.

Oracle Restart

Oracle Restart implements a high availability solution for stand-alone Oracle databases.

- Can monitor and restart the following components:
  - Database instances
  - Oracle Net listener
  - Database services
  - Automatic Storage Management (ASM) instance
  - ASM disk groups
  - Oracle Notification Services (ONS/eONS)
- Runs periodic check operations to monitor the health of the components
- Runs out of the Oracle Grid Infrastructure home, which you install separately from Oracle Database homes

ORACLE

Oracle Restart

Oracle Restart is designed to improve the availability of your Oracle Database. It implements a high availability solution for single instance (nonclustered) environments only. For Oracle Real Application Cluster (Oracle RAC) environments, the functionality to automatically restart components is provided by Oracle Clusterware. Oracle Restart can monitor the health and automatically restart the following components:
- Database instances
- Oracle Net listener
- Database services
- ASM instance
- ASM disk groups
- Oracle Notification Services (ONS/eONS): Service for sending Fast Application Notification (FAN) events to integrated clients upon failover. The eONS is used by Oracle Enterprise Manager to receive notification of change in status of components managed by Oracle Restart.

Restarting an ASM disk group means mounting it. The ability to restart ONS is applicable only in Oracle Data Guard installations for automatic failover of connections between primary and standby databases through FAN.
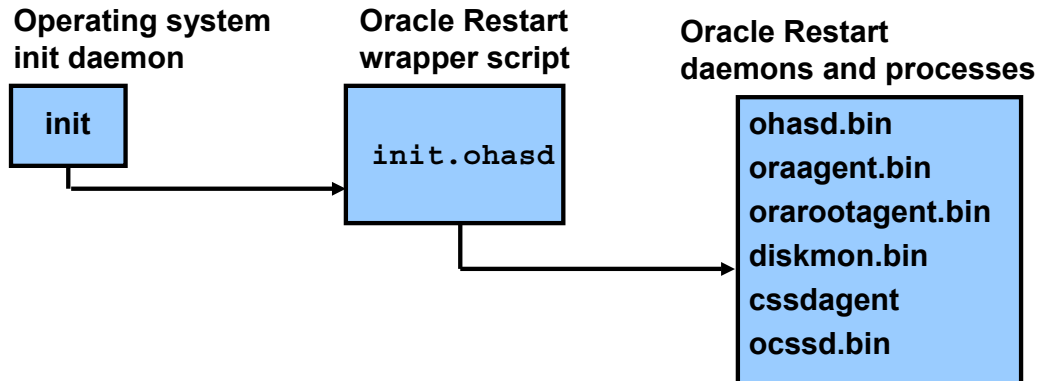
Oracle Restart (continued)

Oracle Restart ensures that the components are started in the proper order, in accordance with component dependencies. If a component must be shut down, it ensures that the dependent components are cleanly shut down first.

Oracle Restart runs out of the Oracle Grid Infrastructure home, which you install separately from Oracle Database homes.

Oracle Restart Process Startup

- Oracle Restart is started by the OS init daemon.

Operating system init daemon	Oracle Restart wrapper script	Oracle Restart daemons and processes
init	**init.ohasd**	**ohasd.bin** **oraagent.bin** **orarootagent.bin** **diskmon.bin** **cssdagent** **ocssd.bin**

- The Oracle Restart installation modifies the `/etc/inittab` file to ensure start up every time the machine starts.

```
# cat /etc/inittab
..
h1:35:respawn:/etc/init.d/init.ohasd run >/dev/null 2>&1 </dev/null
```

ORACLE

Oracle Restart Process Startup

During the installation of Oracle Restart, entries to start a wrapper script are placed in the `/etc/inittab` operating system file. The wrapper script sets up the environment variables and then starts the Oracle Restart daemons and processes.

When a command is used to stop Oracle Restart, the daemons will be stopped, but the wrapper script process will remain running.

The format of the UNIX /etc/inittab file is as follows:

    id : run levels : action : process with parameters

The wrapper script is started with the respawn action so it will be restarted whenever it is terminated. In addition, the respawn action causes the init process to restart the daemons if they fail.

Some of the Oracle Restart daemons will be running under the root user with real-time priority, and others will be running under the Grid Infrastructure owner with user-mode priorities after they are started. On a Windows platform, operating system services are used instead of wrapper initialization scripts and the daemons are executable binaries.

Note: Executing the wrapper script directly is not supported.

Controlling Oracle Restart

The CRSCTL utility can be used to control the state of Oracle Restart.

- To display the Oracle Restart configuration:

```
$ crsctl config has
```

- To enable or disable the automatic restart of Oracle Restart:

```
$ crsctl [ enable | disable ] has
```

- To start or stop Oracle Restart:

```
$ crsctl [ start | stop ] has
```

ORACLE

Controlling Oracle Restart

The CRSCTL utility can be used to control the state of Oracle Restart. It can be used to determine if the autostart capability is enabled or disabled as follows:

```
$ crsctl config has
CRS-4622: Oracle High Availability Services autostart is
enabled.
```

During the installation of Oracle Grid Infrastructure, entries are placed in the /etc/inittab operating system file to start a wrapper script as follows:

```
h1:35:respawn:/etc/init.d/init.ohasd run >/dev/null 2>&1
</dev/null
```

The wrapper script is responsible for setting up environment variables and then starting the Oracle High Availability Services Daemon (ohas) along with other related processes. The wrapper script is executed as the root user.

When the CRSCTL utility is used to disable the automatic restart of the Oracle Grid Infrastructure, the entry in the /etc/inittab file is not removed. The control files for ohas are used to control the state of ohas and determine if automatic restart is enabled or disabled. These are known as SCLS_SCR files. For Linux, the location of the control files is defined to be:

```
/etc/oracle/scls_scr/$HOST/<Oracle Restart owner> and
/etc/oracle/scls_scr/$HOST/root
```

Controlling Oracle Restart (continued)

If the CRSCTL utility is used to stop Oracle Restart, all components currently managed by Oracle Restart will also be stopped.

```
$ crsctl stop has
CRS-4549: Stopping resources.
CRS-2673: Attempting to stop 'ora.diskmon' on 'host01'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host01'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host01'
CRS-2677: Stop of 'ora.DATA.dg' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host01'
CRS-2675: Stop of 'ora.diskmon' on 'host01' succeeded
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host01' succeeded
CRS-2677: Stop of 'ora.asm' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host01'
CRS-2677: Stop of 'ora.cssd' on 'host01' succeeded
CRS-2673: Attempting to stop 'ora.diskmon' on 'host01'
CRS-2677: Stop of 'ora.diskmon' on 'host01' succeeded
CRS-4133: Oracle High Availability Services has been stopped.
```

In the example above, there is no Oracle Database software installed. If Oracle databases had been registered with Oracle Restart or they had been using ASM, they would have been stopped also.

When starting Oracle Restart with the CRSCTL utility, each component that is started is not displayed to standard output.

```
$ crsctl start has
CRS-4123: Oracle High Availability Services has been started.
```

Note: Invoking the wrapper script directly to start the Oracle Grid Infrastructure processes is not supported.

Choosing the Correct SRVCTL Utility

- Invoke the Server Control (SRVCTL) utility from the Oracle Grid Infrastructure home when working with ASM instances, disk groups, listeners, and ONS.

```
$ export ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid
$ $ORACLE_HOME/bin/srvctl command component options
```

- Invoke the SRVCTL utility from the Oracle Database home when working with the database or database instance.

```
$ export
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
$ $ORACLE_HOME/bin/srvctl command component options
```

Choosing the Correct SRVCTL Utility

Oracle Restart includes the SRVCTL utility that you use to start, stop, and manage Oracle Restart components. After the Oracle Database software is installed in addition to the Oracle Grid Infrastructure software, there will be a copy of the SRVCTL utility in each ORACLE_HOME location. You need to determine the correct ORACLE_HOME location in which to run the SRVCTL utility. You need to run the SRVCTL utility from the Grid Infrastructure software home directory when managing the ASM instance, ASM disk groups, Oracle Net listeners, and the ONS. You need to run the SRVCTL utility from the Oracle Database software home directory when managing the Oracle database instances. To determine the currently mapped location of the SRVCTL utility, use the which command as follows:

```
$ which srvctl
/u01/app/oracle/product/11.2.0/grid/bin/srvctl
```

Note: For the Oracle Net listener, the assumption is that Oracle Grid Infrastructure was installed before the Oracle Database software. If Oracle Restart is added to an existing Oracle Database installation, the Oracle Net listener could be running from the Oracle Database home directory. In that case, you should use the SRVCTL utility from the Oracle Database home to manage the Oracle Net listener.

Oracle Restart Configuration

Oracle utilities will automatically update the Oracle Restart configuration.

Create operations and the Oracle Restart configuration	Automatically added to configuration?
Create a database with OUI or DBCA	YES
Create a database with SQL statement	NO
Create an ASM instance with OUI, DBCA, or ASMCA	YES
Create a disk group (any method)	YES
Add a listener with NETCA	YES
Create a database service with SRVCTL	YES
Create a database service by modifying `SERVICE_NAMES` initialization parameter	NO
Create a database service with `DBMS_SERVICE.CREATE SERVICE`	NO

ORACLE

Oracle Restart Configuration

Oracle Restart maintains a list of all the components that it manages, and maintains configuration information for each component. All of this information is collectively known as the "Oracle Restart configuration." When Oracle Restart is installed, many operations that create Oracle components using Oracle utilities will automatically add the components to the Oracle Restart configuration. If a component is created manually without using an Oracle utility, then SRVCTL commands can be used to add it to the Oracle Restart configuration if desired. The table in the above slide shows which create operations automatically add the component to the Oracle Restart configuration and which create operations do not update the Oracle Restart configuration.

Note: The same principal applies to delete operations.

Using the SRVCTL Utility

- The SRVCTL utility is used to start, stop, and manage Oracle Restart components with the following syntax:

```
$ srvctl command component options
```

- The following command and components are supported:

Commands	add config disable enable getenv modify remove setenv start status stop unsetenv
Components	asm db dg filesystem home lsnr serv ons eons

Using the SRVCTL Utility

When Oracle Restart is in use, Oracle strongly recommends that you use the SRVCTL utility to start, stop, and manage all Oracle Restart components. The SRVCTL utility is recommended for the following reasons:

- All dependencies between components are maintained. This enables Oracle Restart to start or to stop any dependent components first.
- Components are started according to their Oracle Restart configuration.
- Environment variables stored in the Oracle Restart configuration for the components are set.

Oracle Restart components can also be started with other utilities such as the listener control (LSNRCTL) utility or SQL*Plus, but the benefits listed above may not be obtained with other utilities. The SRVCTL utility with Oracle Restart supports 12 commands and 9 components. The options that are allowed vary with each command and component combination. The SRVCTL utility syntax is as follows:

```
srvctl command component options
```

where:

- *command* is a verb such as start, stop, or remove
- *component* is the object on which SRVCTL performs the command, such as a database
- *options* extend the use of the preceding command to include additional parameters

Obtaining Help for the SRVCTL Utility

The SRVCTL utility provides online help for its commands, components, and options.

- For help with general usage:

```
$ srvctl -h
```

- For help on a particular command:

```
$ srvctl command -h
```

- For help on a particular command and component:

```
$ srvctl command component -h
```

Obtaining Help for the SRVCTL Utility

The SRVCTL utility provides detailed online help for its commands, components, and options. To display the online Help, you use the help (-h) option to display usage information. If the help (-h) option is the only parameter specified, SRVCTL displays a general outline of all commands with the most common options used for each command and component combination. This will not be a complete list of all supported options. For more detailed and complete information, the help (-h) option can be used for a specific command, or for a specific command and component combination.

Starting Components by Using the SRVCTL Utility

Oracle recommends that the SRVCTL utility be used to start all components.

- Examples of starting individual components:

```
$ srvctl start database –d PROD –o mount
$ srvctl start listener –l crmlistener
$ srvctl start service –d PROD –s "service1,service2"
$ srvctl start diskgroup –g "DATA,FRA"
$ srvctl start asm
$ srvctl start eons -v
$ srvctl start ons
```

- Example of starting all Oracle Restart components in a specified Oracle home:

```
$ srvctl start home –o oracle_home –s state_file
```

ORACLE

Starting Components by Using the SRVCTL Utility

The SRVCTL utility can be used to start individual components, along with any dependent components that are necessary. For example, the `srvctl start database –d PROD` command may also start the listener, the ASM instance, and multiple disk groups if those components have been defined as being managed by Oracle Restart and are listed as dependent components to the `PROD` database.

The SRVCTL utility can also be used to start all components that are associated with a specified Oracle home and have been configured for Oracle Restart with the following command:

```
srvctl start home –o /u01/app/oracle/product/11.2.0/dbhome_1
 –s /usr/local/bin/group_state_file
```

The state file contains the current state information for the components in the Oracle home and is created when the `srvctl status home` command is executed. It is indicated with the state file option (`-s`), and must specify the complete path of the state file. The state file can be created in any directory.

Note: The options shown in the slide represent the most common options and are not a complete list. You can use the help option (`-h`) for a complete list of all available options for each command.

Stopping Components by Using the SRVCTL Utility

Oracle recommends that the SRVCTL utility be used to stop all components.

- Examples of stopping individual components:

```
$ srvctl stop database –d PROD –o transactional
$ srvctl stop listener –l crmlistener -f
$ srvctl stop service –d PROD –s "service1,service2"
$ srvctl stop diskgroup –g "DATA,FRA" -f
$ srvctl stop asm –o immediate -f
$ srvctl stop eons -v
$ srvctl stop ons
```

- Example of stopping all Oracle Restart components in a specified Oracle home:

```
$ srvctl stop home –o oracle_home –s state_file -f
```

ORACLE

Stopping Components by Using the SRVCTL Utility

The SRVCTL utility can be used to stop individual components, along with any dependent components that must be stopped. For example, the `srvctl stop diskgroup –g "DATA -f"` command will force an unmount of the diskgroup even if files are open in it. It will also stop all database instances that depend on the `DATA` disk group.

The SRVCTL utility can also be used to stop all components that are associated with a specified Oracle home and have been configured for Oracle Restart with the following command:

```
srvctl stop home –o /u01/app/oracle/product/11.2.0/dbhome_1 –
s /usr/local/bin/group_state_file -f
```

This can be very useful when it is necessary to stop all components such as when you need to apply a patch to the software binaries.

Note: The options shown in the slide represent the most common options and are not a complete list. You can use the help option (`-h`) for a complete list of all available options for each command.

Viewing Component Status

- You can use the `status` command to view the running status for any component managed by Oracle Restart.
- Display the running status for a database:

```
$ srvctl status database -d orcl
Database is running.
```

- Display the listener status:

```
$ srvctl status lsnr
Listener LISTENER is enabled
Listener LISTENER is running on node(s): host01
```

ORACLE

Viewing Component Status

You can use the SRVCTL utility to view the running status (running or not running) for any component managed by Oracle Restart. Additional information is displayed for some components.

The format of the command is as follows:

```
srvctl status object [options]
```

`object` can be one of the following:
- `asm`: ASM instance
- `db`: Database instance
- `dg`: Oracle ASM disk group
- `filesystem`: Oracle ASM file system
- `home`: Oracle home or Oracle Clusterware home
- `lsnr`: Oracle Net listener
- `ons, eons`: Oracle Notification Services
- `serv`: Database service

Refer to the *Oracle Database Administrators Guide* for a list of options for each of the above objects.

Displaying the Oracle Restart Configuration for a Component

- You can use the `config` command to display the Oracle Restart configuration of a component.
- Display the Oracle Restart configuration for a database:

```
$ srvctl config database –d orcl
Database unique name: orcl
Database name: orcl
Oracle home: /u01/app/oracle/product/11.2.0/dbhome_1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Domain: example.com
Start options: open
Stop options: immediate
Database role:
Management policy: automatic
Disk Groups: DATA,FRA
Services: east,sales
```

ORACLE

Displaying the Oracle Restart Configuration for a Component

You can use the SRVCTL utility to display the Oracle Restart configuration of a component with the `config` command. The `config` command is valid for the `database`, `service`, `asm`, `listener`, `ons`, and `eons` components.

The configuration for an Oracle Restart component can be modified with the SRVCTL utility `modify` command. The following syntax shows an example of how to modify the database with the unique name of `orcl` to use a different, nonstandard directory for the server parameter file (SPFILE).

```
srvctl modify database –d orcl –p
/usr/local/oracle/spfileorcl.ora
```

Manually Adding Components to the Oracle Restart Configuration

Components can be manually added to the Oracle Restart configuration with the `add` command.

- To define a new listener that was not created with NETCA, use the following syntax:

```
$ srvctl add listener –l MYLISTENER -p TCP:1525 –o
/u01/app/oracle/product/11.2.0/grid
```

- To specify a nondefault location for the new listener's networking files, use the following syntax:

```
$ srvctl setenv listener –l MYLISTENER –t
TNS_ADMIN=/usr/local/oracle
```

Manually Adding Components to the Oracle Restart Configuration

The SRVCTL utility can be used to manually add components to the Oracle Restart configuration with the `add` command. If the component was created with an Oracle utility such as NETCA, DBCA, ASMCA, or OUI, it would have been automatically added to the Oracle Restart configuration and it would not be necessary to manually add it.

The slide illustrates manually adding a new listener called `MYLISTENER` to the Oracle Restart configuration. The listener will use the software binaries of the Grid home installation, but will depend on a nonstandard location for the networking files. For this example, it is assumed that the `listener.ora` networking file has been created in the `/usr/local/oracle` directory. The `setenv` command of the `srvctl` utility is used to define environment variables that may be needed for specific components. The `TNS_ADMIN` environment variable is set to the nondefault location of the `listener.ora` file, and is defined only for the listener named `MYLISTENER`. This will not have an impact on any other listeners that may already exist and that use different directories for the networking files.

The SRVCTL utility includes a `remove` command to manually delete a component from the Oracle Restart configuration. To delete the listener created above, use the following syntax:

```
            srvctl remove listener –l mylistener –f
```

This will also remove the environment variable that was associated with the listener.

Quiz

You invoke the SRVCTL utility from the Oracle Grid Infrastructure home when working with:

1. Listeners
2. ASM instances
3. Database instances
4. ASM disk groups

Answers: 1, 2, and 4

Summary

In this lesson, you should have learned how to use Oracle Restart to manage components.

Practice 3-1: Overview

In this practice, you use Oracle Restart to manage components.