Server Patching - SOP

The following outlines the steps to be followed in order to carry out Operating System patching of an Azure database server configured as part of a Data Guard Primary/Standby database pair. There may also be a DR server and database to be patched.

The process outlined should allow patching to be carried out with minimal downtime for the applications and databases, however, some downtime is inevitable.

# Important – RMAN Backup Tasks

When patching is in progress, or has completed, the database server, whichever one it happens to be, running as the primary server, must have the Windows Task Scheduler's RMAN Backup Tasks enabled. These will run at pre-determined times of the day to carry out RMAN backups of the various databases.

The servers running as standby or DR during and after patching, must have their task scheduler backup tasks disabled. We only run the backups on the primary servers in production and pre-production.

# Patch the Current STANDBY Server

As the current PRIMARY is in use for the applications, the STANDBY server should be patched first.

## Disable the STANDBY Processing

Log in to the PRIMARY database and disable log shipping to the standby. The following will generate a command that must be executed on the primary database to prevent updates being applied to the current standby:

select 'alter system set ' ||

replace(dest\_name,'DEST\_','DEST\_STATE\_') ||

'=''DEFER'' scope=memory;'

from v$archive\_dest\_status

where db\_unique\_name = 'STANDBY\_DATABASE\_NAME\_UPPER\_CASE';

The command generated will resemble the following:

alter system set LOG\_ARCHIVE\_DEST\_STATE\_2='DEFER' scope=memory;

You should leave any other log destinations alone. These would usually include the DR server at a separate location, and would normally be on LOG\_ARCHIVE\_DEST\_3.

Log into the STANDBY database and cancel managed recovery:

alter database recover managed standby database cancel;

Shutdown the STANDBY database:

shutdown immediate

exit

## Disable the STANDBY Listener

lsnrctl stop

## Patch the STANDBY Server

Patch the server. Rebooting as often as required! Frequently.

## Restart the STANDBY listener

lsnrctl start

## Startup the STANDBY database

You may need to start the service after a server reboot:

net start OracleService<StandbyDB>

You can ignore errors if it reports the service as already running. The database can now be started.

sqlplus sys/password as sysdba

startup MOUNT

## Reenable log shipping at the PRIMARY database

Login to the PRIMARY database and enable log shipping:

alter system set log\_archive\_dest\_state\_2='ENABLE' scope=memory;

## Startup the STANDBY database

startup MOUNT

The STANDBY database should now automatically start fetching and applying archived logs from the PRIMARY database without any further input from the DBA. However, this *must* be checked, first on the PRIMARY database:

select gap\_status from v$archive\_dest\_status

where dest\_id = 2;

GAP\_STATUS

------------------------

NO GAP

Optionally, on the STANDBY server. Locate the alert log, which will be found in c:\OracleDatabase\diag\rdbms\<standbydb>\<standbydb>\trace\alert\*.log and open it in Notepad++ (or, Notepad, if you really must!)

Go to the end of the file and search backwards for the following text:

Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE THROUGH ALL SWITCHOVER DISCONNECT USING CURRENT LOGFILE

If not found, *and* the PRIMARY showed that a GAP existed, even after a few minutes waiting, you should manually restart managed recovery:

alter database recover managed standby database

using current logfile disconnect;

# Switchover to the Current STANDBY Database

Now that the STANDBY database is up and running on a patched server, it needs to be running as the PRIMARY in order that the current PRIMARY server can be patched.

Switchover the databases so that the current STANDBY becomes the new PRIMARY. This will incur a small downtime, so the applications should be logged out of for the duration of the switchover.

On either the PRIMARY or STANDBY server, use dgmgrl to facilitate the switchover as described in the document "*AZURE - Standby Switchover 11g Standard Operating Procedures*" which can be found in the same TFS location as this document.

In summary:

Set the Oracle environment to the appropriate database (depending on which server you are on) and log into dgmgrl as the sys user, with a password:

dgmgrl sys/password

Check the configuration currently running:

show configuration

Switchover to the listed standby database:

switchover to current\_standby

There are, however, a number of advisable checks that should be carried out first, these are detailed in the above mentioned document.

# Patch the Current PRIMARY Server

This is exactly the same process as patching the previously running STANDBY server. The database and applications should be running on the other server by now, so the patching process can begin.

## Disable the *New* STANDBY Processing

Log in to the *new* PRIMARY database and disable log shipping to the standby. The following will generate a command that must be executed on the new primary database to prevent updates being applied to the current standby:

select 'alter system set '||

replace(dest\_name,'DEST\_','DEST\_STATE\_')||

'=''DEFER'' scope=memory;'

from v$archive\_dest\_status

where db\_unique\_name = 'OLD\_PRIMARY\_DATABASE\_NAME\_UPPER\_CASE';

The command generated will resemble the following:

alter system set LOG\_ARCHIVE\_DEST\_STATE\_2='DEFER' scope=memory;

You should leave any other log destinations alone. These would usually include the DR server as a separate target location, and would normally be on LOG\_ARCHIVE\_DEST\_3.

Log into the *new* STANDBY database and cancel managed recovery:

alter database recover managed standby database cancel;

Shutdown the *new* STANDBY database:

shutdown immediate

exit

## Disable the *New* STANDBY Listener

Shutdown the listener:

lsnrctl stop

## Patch the *New* STANDBY Server

Patch the server. Rebooting as often as required! Frequently.

## Restart the *New* STANDBY listener

lsnrctl start

## Startup the *New* STANDBY database

You may need to start the service after a server reboot:

net start OracleService<NewStandbyDB>

You can ignore errors if it reports the service as already running. The database can now be started.

startup MOUNT

## Reenable log shipping at the *New* PRIMARY database

Login to the *new* PRIMARY database and enable log shipping:

alter system set log\_archive\_dest\_state\_2='ENABLE' scope=memory;

## Startup the *New* STANDBY database

startup MOUNT

The *new* STANDBY database should now automatically start fetching and applying archived logs from the *new* PRIMARY database without any further input from the DBA. However, this *must* be checked, first on the *new* PRIMARY database:

select gap\_status from v$archive\_dest\_status

where dest\_id = 2;

GAP\_STATUS

------------------------

NO GAP

Optionally, on the *new* STANDBY server. Locate the alert log, which will be found in c:\OracleDatabase\diag\rdbms\<standbydb>\<standbydb>\trace\alert\*.log and open it in Notepad++ (or, Notepad, if you really must!)

Go to the end of the file and search backwards for the following text:

Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE THROUGH ALL SWITCHOVER DISCONNECT USING CURRENT LOGFILE

If not found, *and* the *new* PRIMARY showed that a GAP existed, even after a few minutes waiting, you should manually restart managed recovery:

alter database recover managed standby database

using current logfile disconnect;

# Patch the Current DR Server

## Disable the DR Processing

Log in to the PRIMARY database and disable log shipping to the standby. The following will generate a command that must be executed on the new primary database to prevent updates being applied to the current standby:

select 'alter system set '||

replace(dest\_name,'DEST\_','DEST\_STATE\_')||

'=''DEFER'' scope=memory;'

from v$archive\_dest\_status

where db\_unique\_name = DR\_DATABASE\_NAME\_UPPER\_CASE';

The command generated will resemble the following:

alter system set LOG\_ARCHIVE\_DEST\_STATE\_3='DEFER' scope=memory;

You should leave any other log destinations alone. These would usually include the normal standby server at a separate location, and would normally be on LOG\_ARCHIVE\_DEST\_2.

Log into the DR database and cancel managed recovery:

alter database recover managed standby database cancel;

Shutdown the DR database:

shutdown immediate

exit

## Disable the DR Listener

lsnrctl stop

## Patch the DR Server

Patch the server. Rebooting as often as required! Frequently.

## Restart the DR listener

lsnrctl start

## Startup the DR database

You may need to start the service after a server reboot:

net start OracleService<NewStandbyDB>

You can ignore errors if it reports the service as already running. The database can now be started.

sqlplus sys/password as sysdba

startup MOUNT

## Reenable log shipping at the PRIMARY database

Login to the PRIMARY database and enable log shipping:

alter system set log\_archive\_dest\_state\_3='ENABLE' scope=memory;

## Startup the DR database

startup MOUNT

The DR database should now automatically start fetching and applying archived logs from the PRIMARY database without any further input from the DBA. However, this *must* be checked, first on the PRIMARY database:

select gap\_status from v$archive\_dest\_status

where dest\_id = 2;

GAP\_STATUS

------------------------

NO GAP

Optionally, on the DR server. Locate the alert log, which will be found in c:\OracleDatabase\diag\rdbms\<drdb>\<drdb>\trace\alert\*.log and open it in Notepad++ (or, Notepad, if you really must!)

Go to the end of the file and search backwards for the following text:

Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE THROUGH ALL SWITCHOVER DISCONNECT USING CURRENT LOGFILE

If not found, *and* the PRIMARY showed that a GAP existed, even after a few minutes waiting, you should manually restart managed recovery:

alter database recover managed standby database

using current logfile disconnect;