



UNIVERSIDAD
POLITECNICA
DE VALENCIA



UNIVERSIDAD POLITÉCNICA DE VALENCIA
ESCUELA TÉCNICA SUPERIOR DE INFORMÁTICA APLICADA

MIGRACIÓN ORACLE 10G A ORACLE ASM RAC 11G

PROYECTO FIN DE CARRERA

Autor
Miguel Ángel Berenguer García

Director
Alberto J. Pérez

Índice de Contenidos

PLANTEAMIENTO DEL PROYECTO	4
CONCEPTOS	5
<i>Base de datos</i>	5
<i>Instancia de Base de datos</i>	5
<i>Datafiles</i>	5
<i>Control files</i>	6
<i>Redo log files</i>	6
<i>Archive Redo log files</i>	6
<i>Automatic Storage Management - ASM</i>	7
<i>Oracle Clusterware</i>	7
<i>Disco de arranque de Clusterware - Voting disk</i>	8
<i>Oracle Cluster Registry - OCR</i>	8
<i>Cluster Verification Utility - CVU</i>	8
<i>Real Application Clusters - RAC</i>	9
<i>Database Upgrade Assistant -DBUA</i>	9
<i>Database Configuration Assistant - DBCA</i>	9
<i>Database Enterprise Manager - EM</i>	9
<i>Oracle Universal Installer - OUI</i>	10
<i>SQL*PLUS</i>	10
<i>Recovery Manager - RMAN</i>	10
<i>oratab</i>	11
<i>tnsnames.ora</i>	11
<i>listener.ora</i>	11
<i>sqlnet.ora</i>	11
<i>Net Configuration Assistant - NETCA</i>	11
¿POR QUÉ SE MIGRA?	13
<i>Caducidad del soporte</i>	13
<i>Nueva versión de la base de datos</i>	13
CONFIGURACIÓN DEL ENTORNO	21
<i>Configuración de la máquina virtual</i>	21
<i>Instalación de la base de datos 10.2.0.4</i>	45
PREMIGRACIÓN	48
<i>Preparación para Clusterware</i>	48
<i>Preparación para la Base de datos y ASM</i>	56
<i>Preparación para RAC</i>	59
<i>Preparación del almacenamiento</i>	62
EJECUCIÓN DE LA MIGRACIÓN	71
<i>Instalar Clusterware 11.1.0.6</i>	71
<i>Instalar los clustered homes 11.1.0.6 para la base de datos y ASM y crear la instancia ASM con cluster 11.1.0.6</i>	82
<i>Upgrade de single instance 10.2.0.4 a single instance 11.1.0.6</i>	95
<i>Cambiar la single instance 11.1.0.6 a ASM con RMAN</i>	135
<i>Convertir la single instance 11.1.0.6 en una base de datos RAC 11.1.0.6</i>	141

ADICIÓN DE UN SEGUNDO NODO AL RAC.....	153
<i>Prerrequisitos y Dependencias</i>	153
<i>Configuración de los Componentes de Red</i>	153
<i>Instalación de Oracle Clusterware</i>	155
<i>Configuración de Oracle Clusterware</i>	157
<i>Instalación del Software de Oracle</i>	159
<i>Adición de la Nueva Instancia</i>	159
PROBLEMAS ENCONTRADOS.....	161
<i>Falta de recursos</i>	161
<i>Fallo en la configuración del virtual host</i>	163
<i>Configuración de ASM en single instance</i>	170
<i>Migración a RAC mediante RCONFIG</i>	171
<i>Múltiples</i>	172
CONCLUSIONES.....	174
BIBLIOGRAFÍA.....	176

Planteamiento del proyecto

En este proyecto se va a realizar un estudio detallado e implementación de todo el procedimiento que conlleva la migración de un sistema de base de datos Oracle 10.2.0.4 a uno Oracle 11.1.0.6. Además de la migración en sí se plantea el problema con otros condicionantes.

Se plantea realizar, además de la migración de la base de datos, un cambio de almacenamiento, inicialmente en file system, a Automatic Storage Management (ASM) y migrar el sistema, configurado en single instance, a Real Application Clusters (RAC).

Para conseguir estos objetivos se configurará un sistema en máquina virtual con Enterprise Linux Redhat 5. En él se instalará y configurará la base de datos inicial y sobre ésta se irá realizando todo el proceso.

Del mismo modo, se aporta información sobre las razones por las que una empresa querría o debería realizar esta migración y las nuevas características que aporta el software final.

Conceptos

En este apartado vamos a explicar o hacer una pequeña introducción a diferentes elementos que se considera de interés para una mejor comprensión del proyecto y para, en un caso hipotético, utilizarlo como referencia si no se recuerda algo en concreto durante la lectura del mismo.

Se hace especial hincapié en conceptos relacionados con Oracle. La forma en la que se recomienda que se organice su software, elementos típicos en sus instalaciones y, en especial, las herramientas que se van a utilizar a lo largo de todo este trabajo.

Base de datos

Una base de datos es una colección de datos tratados como una unidad. El objetivo de una base de datos es almacenar y recuperar información relacionada. En general, un servidor gestiona de forma fiable una gran cantidad de datos en un entorno multiusuario en el cual muchos de estos pueden acceder simultáneamente a los mismos datos. Todo esto es cumplido mientras se proporciona un rendimiento de alto nivel. Además, una base de datos proporciona también prevención de accesos no autorizados y soluciones eficientes para errores de recuperación.

Instancia de Base de datos

La base de datos Oracle está compuesta por un conjunto de archivos del sistema operativo que contienen datos introducidos por usuarios o aplicaciones e información estructural acerca de la base de datos en sí misma llamados metadata. La información es almacenada de forma persistente en estos archivos.

Para permitir a los usuarios o aplicaciones ver o actualizar los datos en la base de datos, Oracle debe levantar un conjunto de procesos, llamados procesos de background, y asignar una cantidad de memoria para ser utilizada durante la operación con la base de datos. Los procesos de background y la memoria asignada por Oracle conjuntamente forman **la instancia**. Una instancia debe ser levantada para leer y escribir información en la base de datos.

Cuando la instancia de la base de datos no está disponible, los datos están a salvo en la base de datos pero no se puede acceder a ellos a través de usuarios ni de aplicaciones.

Las propiedades de una instancia de base de datos están especificadas utilizando los parámetros de inicialización de la instancia. Cuando la instancia es levantada, un archivo de parámetros de inicialización es leído y la instancia es configurada de acuerdo a éste.

Datafiles

Todas las bases de datos Oracle tienen al menos un datafile físico. Los datafiles contienen todos los datos de la base de datos. Los datos de las estructuras lógicas de

la base de datos, tales como las tablas o los índices, están físicamente almacenados en datafiles asignados para la base de datos.

Las características de los datafiles son:

- Un datafile sólo puede ser asociado con una única base de datos.
- Los datafiles pueden tener ciertas características para que automáticamente se extiendan cuando la base de datos se quede sin espacio.
- Uno o más datafiles forman una unidad lógica de almacenamiento llamada tablespace.

Control files

Todas las bases de datos Oracle tiene al menos un control file. Un control file contiene entradas que especifican la estructura física de la base de datos.

Los control files contienen la siguiente información:

- El nombre de la base de datos
- El nombre y la localización de los datafiles y los redo log files
- Fecha de creación de la base de datos

Oracle puede multiplexar el control file, esto es, simultáneamente mantener un número de copias idénticas del control file, para protegerlo contra un fallo que pudiera afectarlo.

Cada vez que una instancia de base de datos Oracle es levantada, su control file identifica la base de datos y redo logs que deben ser levantados para que la base de datos pueda operar. Si la configuración física de la base de datos es alterada (por ejemplo, si se le añade un nuevo datafile o redo log), el control file es automáticamente modificado por Oracle para reflejar el cambio.

Redo log files

Los archivos redo log son la estructura crucial en las operaciones de recovery o recuperación de la base de datos. El redo log consiste en dos o más archivos preasignados que almacenan todos los cambios hechos en la base de datos tal como ocurren. Toda base de datos Oracle tiene asociado un redo log para proteger la misma en caso de un fallo de la instancia.

Un archivo de redo log está formado por entradas de redo o redo records y su función primaria es registrar todos los cambios hechos sobre los datos. Del mismo modo, para proteger al sistema de un fallo que afectara al redo log en sí mismo, Oracle multiplexa los archivos de redo log y es por eso por lo que siempre habrá al menos dos copias.

Archive Redo log files

Cuando se archiva el redo log, se escribe los archivos de redo a otra localización antes de ser sobrescritos. Esta localización es llamada el **archive log**. Se puede archivar a múltiples localizaciones, incluyendo una base de datos standby.

Estas copias de los archivos de redo extienden el conjunto de datos de redo que pueden ser salvados para ser utilizados en una operación de recuperación. El archivado puede ser habilitado o no para la base de datos, pero Oracle recomienda que esté en modo ARCHIVELOG.

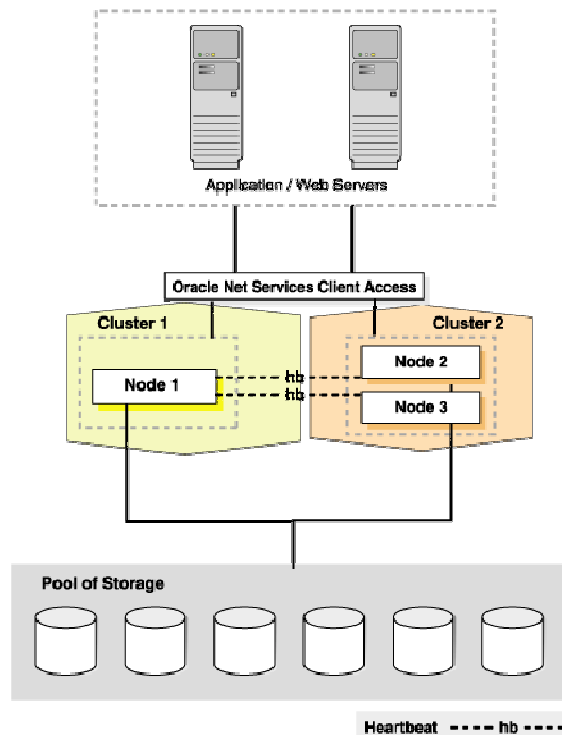
Automatic Storage Management - ASM

ASM automatiza y simplifica la distribución de los datafiles, control files y log files. Los ficheros de la BD son automáticamente distribuidos por todos los discos disponibles, y el almacenamiento de la BD es rebalanceado siempre que la configuración del almacenamiento cambia. Proporciona redundancia a través de mirroring (espejamiento o duplicación) de los ficheros de la BD, e incrementa su rendimiento mediante la automatización de la distribución de los archivos de la BD por todos los discos disponibles.

Oracle Clusterware

Oracle Clusterware es un software que permite a los servidores trabajar juntos como si fueran uno solo. Cada servidor parece un servidor standalone, sin embargo, cada uno tiene unos procesos adicionales para comunicarse con los otros por lo que el conjunto de los servidores aparece como un solo servidor para las aplicaciones y los usuarios.

La siguiente figura muestra una configuración que utiliza Oracle Clusterware para extender la arquitectura básica de una single-instance de base de datos. En la figura, los dos clusters están conectados a la base de datos y están sirviendo activamente a los usuarios y las aplicaciones.



Los beneficios de utilizar un cluster incluyen:

- Escalabilidad para las aplicaciones
- Utilización de un hardware de menos coste
- Habilidad para hacer crecer la capacidad a través del tiempo añadiendo servidores cuando sea necesario

Se puede programar Oracle Clusterware para administrar la disponibilidad de las aplicaciones de usuario y de las bases de datos. En un entorno RAC, Clusterware administra todos los procesos de base de datos automáticamente. Todo lo que es administrado por el Clusterware es conocido como un cluster resource, puede ser una base de datos, una instancia, un servicio, un listener,...

Disco de arranque de Clusterware - Voting disk

Oracle Clusterware utiliza el disco de arranque para determinar qué instancias son miembros del cluster y debe residir en un disco compartido. Para conseguir alta disponibilidad, Oracle recomienda tener un mínimo de tres voting disks. Si se configura un solo voting disk, como es nuestro caso, se debe utilizar redundancia externa. Se puede tener hasta 32 voting disks en un mismo cluster.

Oracle Cluster Registry - OCR

Oracle Clusterware utiliza OCR para almacenar y administrar información acerca de los componentes que Clusterware controla, tales como las bases de datos RAC, listeners, VIPs, servicios y cualquier aplicación. El repositorio de OCR almacena información de configuración en series de pares clave-valor en un directorio con estructura de árbol.

Oracle recomienda utilizar OCR multiplexado para asegurar alta disponibilidad del cluster. Las características de OCR son:

- El OCR debe residir en un disco compartido al que tengan acceso todos los nodos del cluster.
- Clusterware puede multiplexar el OCR.
- Se puede sustituir un OCR fallido online.
- Se debe actualizar el OCR mediante APIs soportadas como Enterprise Manager, Server Control Utility (srvctl) o database configuration Assistant (dbca).
- Clusterware necesita que cada nodo esté conectado a una red privada a través de un interconnect privado. Para obtener redundancia se puede tener hasta 32 voting disks y una ventana del OCR.

Cluster Verification Utility - CVU

El Cluster Verification Utility (CVU) es una herramienta para realizar chequeos de sistema para preparar instalaciones, actualizaciones de parches u otros cambios en el sistema.

El CVU puede verificar los componentes primarios del cluster durante una fase o escenario operacional. Un componente puede ser básico, como espacio de disco

libre, o puede ser complejo, como el chequeo de la integridad del Clusterware. Por ejemplo, CVU puede verificar múltiples subcomponentes del Clusterware a través de todas las capas del Clusterware. Adicionalmente, CVU puede chequear el espacio de disco, la memoria, procesos y otros componentes importantes del cluster. Un escenario podría ser, por ejemplo, la instalación de una base de datos, para la cual CVU puede verificar si el sistema cumple con todos los criterios para la instalación de un RAC.

Real Application Clusters - RAC

Las bases de datos RAC son sistemas de alta disponibilidad. Los clusters que son típicos de entornos RAC pueden proveer servicios continuos durante apagones planeados y no planeados. RAC construye altos niveles de disponibilidad en las más importantes características de Oracle. Todas las características de alta disponibilidad de una single instance, tales como recoveries rápidos y reorganización online, se aplica también a RAC.

Además de todas las características regulares de Oracle, RAC explota la redundancia proporcionada por la clusterización para conseguir disponibilidad en un cluster de n nodos con fallo en n-1 nodos. En otras palabras, todos los usuarios tienen acceso a todos los datos mientras que se mantiene online un nodo del cluster.

Database Upgrade Assistant -DBUA

Database Upgrade Assistant, DBUA, guía a través de todo el proceso de upgrade y configura la base de datos para la nueva versión. DBUA automatiza el proceso de upgrade y hace las recomendaciones apropiadas para las opciones de configuración como tablespaces y redo logs.

DBUA proporciona soporte para Real Application Clusters (RAC) y Automatic Storage Management (ASM).

Database Configuration Assistant - DBCA

Database Configuration Assistant, DBCA, crea una base de datos a partir de plantillas proporcionadas por Oracle, o puede crear una base de datos con las configuraciones que indique el usuario. Permite también copiar una base de datos preconfigurada, que salve el tiempo y el esfuerzo de generar y customizar una nueva desde cero.

Database Enterprise Manager - EM

Oracle Enterprise Manager es la herramienta primaria para la administración de la base de datos Oracle. Tiene una interfaz web. Después de instalar el software de Oracle, creada o actualizada la base de datos, y configurada la red, se puede utilizar EM para administrar la base de datos. Además, EM proporciona también una

interfaz de asesor de rendimiento y otras utilidades como SQL*Loader y Recovery Manager.

Oracle Universal Installer - OUI

Oracle Universal Installer es el encargado de instalar el software de oracle y sus opciones. Puede ejecutar automáticamente DBCA para instalar una base de datos y otros.

SQL*PLUS

SQL*Plus es una herramienta interactiva y de consulta por lotes que es instalada con todos los servidores de base de datos Oracle. Tiene un interfaz de usuario mediante línea de comandos, una interfaz gráfica y la iSQL*Plus web-based user interface.

SQL*Plus tiene sus propios comandos y entorno, y proporciona acceso a la base de datos. Permite al usuario acceder y ejecutar SQL, PL/SQL, SQL*Plus y comandos del sistema operativo para realizar lo siguiente:

- Formatear, realizar cálculos, almacenar e imprimir el resultado de queries
- Examinar definiciones de tablas y objetos
- Desarrollar y ejecutar batch scripts
- Realizar la administración de la base de datos

Se puede utilizar SQL*Plus para generar informes interactivamente, para generar informes como procesos batch, y para sacar los resultados a un archivo de text, a la pantalla, o a un archivo HTML para mostrarlo en Internet.

Recovery Manager - RMAN

Oracle Recovery Manager, RMAN, es una herramienta que puede ser utilizada via Enterprise Manager y via comandos y es el método preferido por Oracle para realizar backups y recoveries de la base de datos eficientemente. RMAN está diseñado para trabajar estrechamente con el servidor, proporcionando detección de corrupciones a nivel de bloque durante el backup o la restauración. RMAN optimiza el rendimiento y el consumo de espacio durante el backup con multiplexación de archivos y compresión de backup.

RMAN tiene en cuenta todos los procesos subyacentes de la base de datos antes y después del backup o restore, liberando la dependencia del sistema operativo y los scripts de sqlplus. Proporciona una interfaz común para las tareas de backup a través de diferentes sistemas operativos de host, y ofrece características no disponibles a través de métodos administrados por usuario como paralelización del backup/recovery de streams de datos, políticas de retención de archivos de backup, e historia detallada de todos los backups.

oratab

El archivo oratab contiene información sobre todas las instalaciones Oracle y todas las bases de datos instaladas en el nodo. Esta información es utilizada para conocer todos los productos Oracle instalados. Cuando un producto Oracle es instalado, este archivo es actualizado con información acerca del nuevo Oracle home. Si este archivo no es encontrado, se crea un nuevo oratab.

tnsnames.ora

tnsnames.ora, situado en `$_ORACLE_HOME/network/admin`, es un archivo de configuración que contiene nombres de servicios de red asignados a descriptores de conexión para el método de local naming, o nombres de servicios de red asignados a direcciones de protocolo de listener.

Un nombre de servicio de red es un alias asignado a una dirección de red de base de datos contenida en un descriptor de conexión. Un descriptor de conexión contiene la localización del listener a través de una dirección de protocolo y el nombre del servicio de la base de datos al que conectarse. Los clientes y servidores de base de datos (que son clientes de otros servidores de bases de datos) utilizan el nombre de servicio de red para realizar la conexión con una aplicación.

listener.ora

listener.ora, situado en `$_ORACLE_HOME/network/admin`, es un archivo que se utiliza para la configuración de Oracle Database Listeners (necesario para aceptar peticiones de conexión remotas).

La configuración de un listener, almacenada en listener.ora, consta de los siguientes elementos:

- Nombre del listener
- Dirección de protocolo donde el listener acepta peticiones
- Servicios de la base de datos
- Parámetros de control

sqlnet.ora

sqlnet.ora es un archivo situado en `ORACLE_HOME/network/admin` que proporciona detalles básicos de configuración como opciones de trazo, dominio por defecto, encriptación...

Net Configuration Assistant - NETCA

Durante una instalación típica de base de datos, Net Configuration Assistant, NETCA, configura automáticamente un listener llamado *LISTENER* que tiene una dirección de protocolo TCP/IP escuchando para la base de datos.

Net Configuration Assistant es una herramienta para configurar manualmente listeners, métodos de nomenclatura, nombres de servicio de red... Oracle recomienda su utilización después de la instalación de la base de datos. Tras esto, se puede administrar la red mediante Enterprise Manager y Oracle Net Manager.

¿Por qué se migra?

Las razones que llevan a migrar una base de datos que está trabajando a pleno rendimiento en todos los entornos se pueden resumir en dos: **caducidad del soporte** y **aparición de nuevas versiones de la base de datos** con nuevas mejoras y funcionalidades que, se sobreentiende, proporcionan una mejora en el rendimiento y mayores funcionalidades.

Caducidad del soporte

Oracle ofrece una serie de servicios para los productos que tiene soportados. Todos estos servicios enfocados a la infraestructura de IT de la empresa que los contrata para soportar las demandas de su negocio.

Todos los productos tienen una caducidad en cuanto a soporte se refiere. Esto hace referencia a varios conceptos. Por un lado se hablaría de las actualizaciones y desarrollo de mejoras del producto contratado y por otro de los servicios de soporte propiamente dichos.

Anualmente se puede hacer actualizaciones o upgrades de los productos contratados mientras se mantiene el soporte. Es decir, Oracle se compromete a trabajar en la mejora constante del producto contratado hasta que aparezca una nueva versión que desoporte la actual.

Mientras se mantenga este soporte, se proporciona el servicio de Resolución de problemas 24 x 7 (7 días - 24 horas) a través de Metalink (procedimiento de contacto por defecto) y a través del teléfono/ email (sólo en ocasiones excepcionales). Estos dos procedimientos son el medio de contacto con los analistas que la empresa dispone para la resolución de los distintos casos que pueden aparecer y que proporcionan una respuesta basada en el estudio del caso concreto y de las mejores prácticas recomendadas.

Una vez haya aparecido una nueva versión de un producto estos servicios de soporte se mantendrán durante un tiempo para que las empresas que lo consideren puedan hacer una transacción. Una vez pasado el período los servicios de actualización de software y de soporte se suspenderán.

Nueva versión de la base de datos

La aparición de nuevas versiones en el software es algo inevitable y con éstas las nuevas funcionalidades y posibilidades para el rendimiento de nuestra base de datos y, en consecuencia, de nuestros sistemas.

En el presente proyecto se presenta la migración de una base de datos Oracle 10g a una Oracle 11g. A continuación mostramos una lista de nuevas funcionalidades que nos presenta esta base de datos y que, por tanto, nos llevan a la consecución de la migración.

Database replay

¿Cuál es la mayor preocupación que tenemos cuando debemos hacer un cambio en la base de datos? Nuestra preocupación más grande es que el cambio “rompa” algo. Un cambio pequeño puede producir un efecto dominó con un impacto importante. Esta funcionalidad permite guardar todas las operaciones de la base de datos – relacionadas y no relacionadas con DML – dentro de la misma base de datos y, reejecutarlas en la misma secuencia que lo hicieron en el caso de producirse un error para recuperar el estado correcto anterior.

Administración de transacciones con LogMiner y Flashback Data Archive

El uso más importante para el que se suele utilizar **LogMiner** es para saber “quién” ha hecho “qué”. Si no se tiene habilitada la auditoría lo único que hay que hacer es buscar la pista de la transacción en la interfaz de LogMiner, en esta versión 11g muy mejorada y sencilla de usar.

La base de datos Oracle 11g combina la utilización de Flashback queries (queries para poder acceder a los datos “pre-cambiados”) con la utilización de Flashback Versions Queries (elementos para seguir la pista a los cambios hechos a una fila mientras los cambios aún están presentes en un segmento de undo). Como resultado se presenta **Flashback Data Archive** que une la simplicidad y la potencia de las Flashback queries pero que no se basa en el almacenamiento transitorio (undo). Ahora se guarda los cambios en una localización permanente, el Flashback Recovery Area.

Cursores adaptables y Plan de gestión SQL

Los cursores adaptables (Adaptive Cursors) y el Plan de gestión SQL (SQL Plan Management) son sólo dos ejemplos de cómo la base de datos es ahora muy inteligente acerca de las peticiones que recibe y de cómo reaccionar a ellas. Con los **cursores adaptables**, podemos usar variables de riesgo y evitar la utilización de un plan sub-óptimo. Y con un **Plan de gestión SQL**, el plan de ejecución no es inamovible, si no que puede evolucionar con el tiempo manteniendo mientras tanto la estabilidad en el corto plazo.

SQL Access Advisor

Considerando un escenario como el siguiente: un índice ayuda sin ninguna duda a una query pero la query va a ser ejecutada sólo una vez. A pesar de que la query se va a beneficiar de la creación del índice, la creación del mismo tendrá un coste que sobrepasará el beneficio. Para analizar el escenario de esa manera, se necesitaría saber con qué frecuencia es accedida la query y por qué.

El Asesor **SQL Access Advisor** realiza este tipo de análisis. Además del análisis de índices, las vistas materializadas y demás que hace Oracle 10g, en Oracle 11g el SQL Access Advisor también analiza tablas y queries para identificar posibles estrategias de particionamiento. Una gran ayuda cuando se diseña un esquema óptimo. SQL Access Advisor puede proporcionar ahora recomendaciones con respecto a la carga de trabajo entera, incluyendo consideraciones sobre el coste de creación y mantenimiento de estructuras de acceso.

RMAN

RMAN, en la versión 11g aporta diferentes novedades. Se podría destacar El **Back up paralelo de un mismo datafile**. Esta nueva funcionalidad permite que se genere varios canales a través de los cuales se haga el back up *paralelizado* pero añadiendo la novedad de que ahora los canales pueden dividirse en trozos, llamados *sections*, el datafile y con ello conseguir una paralelización más *real*.

Flashback logs para recuperar es también una nueva funcionalidad interesante. Hasta la versión 10g se guardaba en logs las versiones óptimas de bloques antes de sufrir cambios generadas en el Flashback Recovery Area. Esto permitía llevar a la base de datos a un punto anterior en el tiempo sin tener que hacer una recuperación desde los backups. A partir de 11g se reutiliza esta técnica para realizar la recuperación utilizando los flashback logs en vez de los backups y ganando así mucho tiempo, sobre todo si los backups están en cintas.

La **Compresión mediante ZLIB** es una nueva posibilidad ofrecida con la cual, es posible comprimir piezas de backup de forma rápida. Es más rápido que su antecesor BZIP2 pero no comprime una gran cantidad. Por otro lado, no consume mucha CPU. La utilización de éste conllevará una mejora en el ancho de banda de la red.

Cuando se tiene múltiples catálogos, se puede crear catálogos virtuales en la misma base de catálogos, por ello se puede ver la necesidad de consolidar todos estos repositorios independientes en uno solo. Con Oracle 11g se puede fusionar los catálogos (**Merging Catalogs**). De hecho, se importa el catálogo de una base de datos a otra.

Por último destacar también la nueva metodología que utiliza RMAN para realizar los **Backup Committed Undo**. En esta nueva 11g, cuando RMAN realiza el backup ignora toda la información y datos *committed undo* que no sean necesarios para la recuperación (estos datos estarán en los datafiles o en los redo log streams). Los datos *uncommitted undo* que sí son importantes para la recuperación son *backed up* normalmente. Esto reduce considerablemente el tamaño y el tiempo invertido en la realización del backup (y de la recuperación igualmente).

Automatic Storage Management

En esta nueva Oracle 11g destacamos, dentro de las novedades para el ASM (Automatic Storage Management), el nuevo **rol SYSASM** para administrar las instancias de ASM solamente, los **tamaños de extent variables** para reducir el uso de la shared pool (utilizada para asignar memoria para ejecución de SQL y PL/SQL) y la habilidad de una instancia para **leer de un disco específico** de un grupo de discos (diskgroup). Todas estas novedades hacen que ASM en 11g sea más potente, adaptable y fácil de usar.

Caching y Pooling

Considerando, por ejemplo, una tabla grande que almacena una lista que es relativamente estática, pero no del todo ya que cambia raramente. Si fuéramos a cachear la tabla de algún modo, ¿cómo podríamos estar seguros de que los datos que estamos cogiendo son correctos cuando algo cambia? La respuesta es **SQL Result Cache**. SQL Result Cache es otra cache dentro del SGA. Cuando se ejecuta una query con la pista *result_cache*, Oracle realiza la operación como cualquier otra pero los resultados son almacenados en el SQL Result Cache. Las subsiguientes invocaciones

de la misma consulta no tendrán que ir a la tabla/s ya que obtendrán los resultados de la cache.

Oracle 11g también permite que se **cachee los resultados de una función PL/SQL** que haga join de varias tablas. Se puede habilitar que los resultados de una función sean cacheados, simplemente poniendo la cláusula `result_cache`.

Oracle 11 ofrece también **Client Query Result Cache**, una herramienta que permite a los clientes cachear los resultados de consultas SQL localmente y no en el servidor, consiguiendo así una mayor escalabilidad para el servidor, consistencia para el RAC y asegurando la eliminación de viajes de ida y vuelta al servidor.

Database Resident Connection Pool (DRCP) es lo que se suele llamar un server-side pool. La funcionalidad de éste es importante porque un sólo pool puede sostener decenas de miles de usuarios activos en una plataforma. Además un sólo pool puede ser compartido a través de múltiples clientes o nodos de middle-tier, y en entornos de RAC y Data Guard, la combinación de DRCP y eventos FAN proporcionan conexión rápida en caso de fallo.

SecureFiles: los nuevos LOBs

La mayor parte de la información es almacenada en la base de datos con un formato relacional pero también es cada vez más probable que se pueda necesitar almacenar datos en una forma no-estructurada o semi-estructurada. Esta información puede ser almacenada de dos formas: mediante campos en la base de datos LOB o mediante ficheros del Sistema Operativo (OS files). Ambos tienen ventajas y desventajas.

En Oracle 11g aparece una nueva infraestructura dentro de la base de datos que proporciona las mejores características de los residentes en base de datos LOBs y los OS files; los **SecureFiles**.

Los SecureFiles no son simplemente la nueva generación de los LOBs. Les añaden mucho más valor, especialmente las características que estuvieron anteriormente en el dominio de los filesystems. Los SecureFiles pueden ser encriptados por seguridad, de-duplicados y comprimidos para obtener un almacenamiento más eficiente, cacheados (o no) para un acceso más rápido, y logueados en diferentes niveles para reducir el tiempo medio necesitado para recuperarse después de una ruptura. Con SecureFiles se puede almacenar más documentos no-estructurados en la base de datos sin incurrir demasiado en gastos indirectos o pérdidas de cualquier funcionalidad crítica alcanzada por un sistema de ficheros del sistema operativo.

Data Guard

La mayor ventaja llega por el hecho de que la base de datos standby puede ahora ser usada productivamente para soportar el negocio en cierta medida. **Active Data Guard** permite abrir la base de datos standby permitiendo a su vez consultas mientras los archived logs están todavía siendo aplicados, es decir, permite estar en sincronía con la base de datos primaria mientras se puede estar utilizando la standby para reporting.

Del mismo modo, la base de datos standby puede convertirse en una **standby Snapshot** para ejecutar cargas de base de datos en producción para, por ejemplo, probar nuevas aplicaciones que afectan a diferentes tablas, comprobar el impacto y luego volver al punto de inicio reiniciando el proceso normal de recuperación.

Estas funcionalidades permiten una mejora de las capacidades de procesamiento disponibles en el servidor de la standby y serían un buen catalizador para una posible subida de versión de la 11g.

Particionando para la perfección

Desde la versión 8 de la base de datos Oracle, siempre se ha podido particionar una tabla o un índice en varios segmentos y localizarlos en tablespaces diferentes. En Oracle 11g aparecen nuevas mejoras referentes al particionamiento tales como el **Interval Partitioning**, que permite que la base de datos cree automáticamente particiones en base a rangos; **Virtual Columns**, que permite que se pueda particionar una tabla sobre una columna virtual que no está almacenada en la tabla; o la extensión del **Composite Partitioning** a rango-rango, lista-rango, lista-hash, y lista-lista, que expone nuevas posibilidades para unas mejores elecciones y manejabilidad del particionamiento.

Administración de esquemas

La administración de los objetos de la base de datos se mejora de manera notoria mediante nuevas funcionalidades que realizan muchas operaciones comunes de una nueva manera increíblemente rápida y simple.

Estas nuevas funcionalidades incluyen **DDL Wait Option**. Esta nueva herramienta permite que cuando se ejecuta una sentencia DDL en una sesión y no consigue obtener el lock exclusivo, no devuelve un error. En vez de esto, permanece durante 10 segundos reintentando ejecutar la operación hasta que lo consigue o el tiempo expira.

La posibilidad de añadir **columnas con un valor por defecto**, es otra nueva opción que ofrece Oracle 11g. Actualmente se podrá definir una nueva columna sobre una tabla ya existente no nula y con un valor por defecto sin incurrir en una actualización que llenara los segmentos de undo, generara una gran cantidad de redo o forzara a un trabajo excesivo. Del mismo modo, mediante la creación de una nueva **columna virtual** se puede dar sentido al negocio sin añadir ninguna complejidad o impacto a la ejecución evitando tener que recurrir a la escritura de triggers u otros.

Oracle 11g incluye toneladas de funcionalidades que no hacen sino simplificar trabajo como las nombradas aquí y otras muchas como los **Índices Invisibles** o la **Tablas de Sólo Lectura**.

Analizador de rendimiento de SQL

Se puede estar interesado en conocer el impacto de ciertos cambios de parámetros en algunas SQLs que aún no están siendo ejecutadas en producción. Aquí aparece el **SQL Performance Analyzer** (Analizador de rendimiento de SQL). Esta herramienta permite aplicar algunas SQLs específicas o el workload entero de SQL contra varios tipos de cambios para luego producir un informe de comparación que nos asesore sobre el impacto.

Codificación eficaz de PL/SQL

En ocasiones, para realizar ciertas operaciones es necesaria la utilización de diferentes códigos (triggers). En 11g estos códigos se pueden unir en un solo **Compound Trigger** para facilitar la programación. Igualmente, se introduce la construcción **continue**, utilizada en bucles, que mueve la lógica hasta el final del bucle y la retorna luego al principio. Ahora se puede introducir una cláusula en el script de creación de los triggers que fuerce la **ordenación de la ejecución** de los mismos. Para alertar de la posibilidad de la aparición de la excepción *others* se ha creado el warning en tiempo de compilación **PLW-06009**. Estas mejoras y muchas otras son las introducidas en Oracle 11g para hacer más fácil la programación PL/SQL.

Seguridad

Oracle 11g ofrece una nueva y rica serie de funcionalidades en el plano de la seguridad. Ofrece una forma muy rápida de identificar usuarios mediante **default passwords** (contraseñas por defecto). También, en las versiones anteriores a 11g las contraseñas de usuario eran casos insensibles. En Oracle 11g pueden ser sensibles también. La posibilidad de forzar una **contraseña** a **case-sensitive** no solo dificulta el crackeo de contraseñas por la fuerza bruta, sino que además permite que se haga cumplir muchos más requisitos de conformidad.

Igualmente se presenta la **Encriptación Transparente de Tablespace**. Esto nos permite por un lado, encriptar los datos almacenados en discos y por otro, hacerlo sin que ello afecte al funcionamiento/rendimiento ya que la administración de los datos se realiza dentro del SGA.

La mejora en la **Auditoría Out-of-box** es también importante. Con el sólo cambio de dos parámetros se consigue una solución para auditoría mucho más potente con un impacto sobre el rendimiento mínimo. El parámetro de base de datos *audit_trail* estará ahora a DB por defecto, en vez de NONE. Esto permite auditar cualquier objeto, sentencia o privilegio sin tener que reciclar la base de datos. El segundo cambio es el aumento de las sentencias que pueden ser auditadas.

Además de las comentadas también se incluyen otras mejoras y funcionalidades como la nueva **Encriptación de los Data Pump DumpFiles**, la **Access Control List** o el nuevo **Data Masking**.

Razonabilidad

Una instancia de base de datos es, resumidamente, una colección de procesos tales como PMON, SMON, y áreas de memoria como System Global Area (SGA) y Program Global Area (PGA). Cada una de estas áreas a su vez tiene subáreas luego, ¿cómo se puede decidir qué tamaño tiene cada una? Hasta Oracle 10g, se debía tomar diferentes decisiones respecto del SGA y el PGA que podían afectar al rendimiento y buen funcionamiento de la base de datos. Ahora, en Oracle 11g, en vez de especificar valores diferentes para cada área, se especifica solamente MEMORY_TARGET simplificando mucho la configuración.

Los nuevos **Umbrales Adaptativos** permiten que los umbrales se adapten a valores específicos basados en el análisis del comportamiento pasado y que están basados en las instantáneas (snapshots) AWR calculadas anteriormente.

En Oracle 11g las estadísticas para una tabla o índice pueden ser definidas como **Pending**, lo que significa que las estadísticas recientemente reunidas no serán

publicadas o usadas por el optimizador – dándonos así una oportunidad de probar las nuevas estadísticas antes de su publicación.

Oracle 11g presenta una nueva funcionalidad realmente revolucionaria, el **Online Patching**. Muchos, aunque no todos los parches, en 11g, específicamente los parches de diagnóstico, pueden ser aplicados online sin perder actividad.

¿No sería interesante poder saber si hay parches disponibles sólo para las funcionalidades que estamos utilizando y así evitarnos las inactividades propias de la aplicación de ciertos parches? Y, todavía más importante, ¿cómo se podría descargar sólo los parches que nos interesan y luego aplicarlos en el momento más conveniente? **Features-Based Patching** en 11g nos permite hacer todas estas cosas.

Pivot y Unpivot

Pivot añade una funcionalidad muy importante y práctica al lenguaje SQL. En vez de escribir complicadas y no-intuitivas secciones de código con montones de funciones descodificadas, se puede utilizar la función *pivot* para crear un informe cruzado sobre cualquier tabla relacional. De forma similar, se puede almacenar cualquier informe cruzado como una tabla relacional regular usando la operación **Unpivot**.

Funcionalidades de recuperación

¿Cómo saber si la base de datos funciona con total corrección? La respuesta es probando “todo”. Con Oracle 11g el esfuerzo se reduce al introducir el **Automatic Health Monitor**, el cual, chequea automáticamente después de un error o bajo demanda, componentes como los datafiles y el diccionario para cerciorarse de que no están corruptos lógicamente o físicamente. Toda esta información, además de todos los eventos críticos, los logs y las trazas, es recogida y guardada para su posterior análisis y procesamiento en el **Automatic Diagnostic Repository (ADR)**.

También el nuevo **Incident Packaging Service** permite hacer un paquete con todas las cuestiones y archivos de soporte para informar fácilmente a Oracle Support.

En Oracle 11g, el **alert log** está escrito en formato XML. Para mantener la compatibilidad con herramientas más antiguas, el alert log tradicional se mantiene también dentro del ADR.

Rendimiento de PL/SQL

Oracle 11g introduce un número de ingeniosas nuevas funcionalidades para mejorar el rendimiento de PL/SQL, pero las más dramáticas son Native compilation e Intra-unit inlining.

Native compilation no es una nueva funcionalidad en sí misma, pero la falta de cualquier precondición para utilizarlo- como la instalación de un compilador de C-es, de hecho, nuevo. Ahora se le llama **Real Native Compilation**. Además, un nuevo tipo de datos, *simple_integer*, hace que el código rinda mejor bajo compilación nativa. **Intra-unit inlining** es una técnica de optimización aplicada al código PL/SQL durante el tiempo de compilación para producir código eficiente.

Configuración del entorno

En este apartado vamos a mostrar cómo se ha configurado todo el entorno en el que se ha desarrollado este proyecto.

A modo de introducción decir que se ha partido con la configuración de un sistema de un solo nodo mediante una máquina virtual con una base de datos Oracle 10.2.0.4 y que se ha ampliado hasta un sistema de dos nodos, ambos con el mismo sistema operativo, con la consiguiente migración de la base de datos a su versión 11.1.0.6, que es el objetivo principal de este proyecto.

Configuración de la máquina virtual

A lo largo de todo este punto se va a mostrar todo el proceso que se ha realizado para la configuración de los nodos, con su sistema operativo y la posterior puesta a punto de la base de datos inicial, sobre la que se realizará la migración.

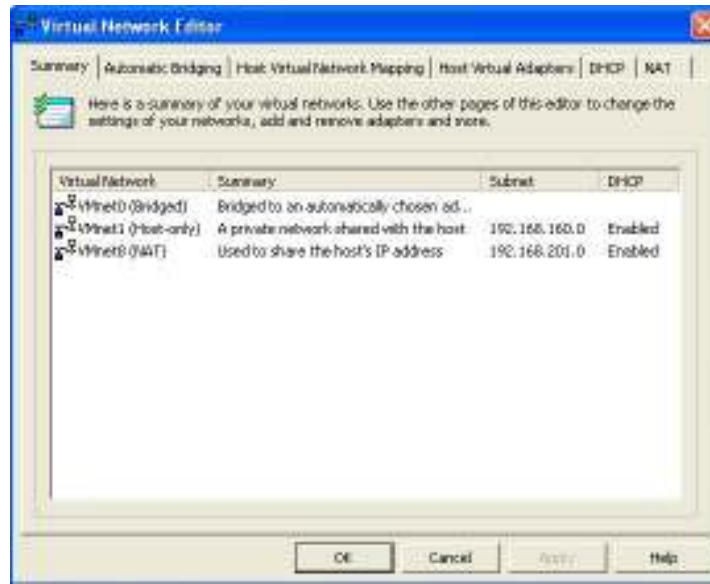
En la parte de las máquinas virtuales nos vamos a centrar especialmente en la configuración de las máquinas virtuales en sí y del sistema operativo ya que algunos temas importantes como la interconexión entre nodos o los dispositivos de almacenamiento son descritos en apartados posteriores.

Siendo ya más concretos decir que se ha trabajado con **Vmware Server 2.0.1** para las máquinas virtuales. El sistema operativo elegido ha sido **Red Hat 5** ya que es un sistema certificado con todos los productos de Oracle. El almacenamiento se ha configurado mediante **discos virtuales** montados en la máquina virtual y las **redes** que se han utilizado ha sido también las **virtuales** habilitadas para la máquina.

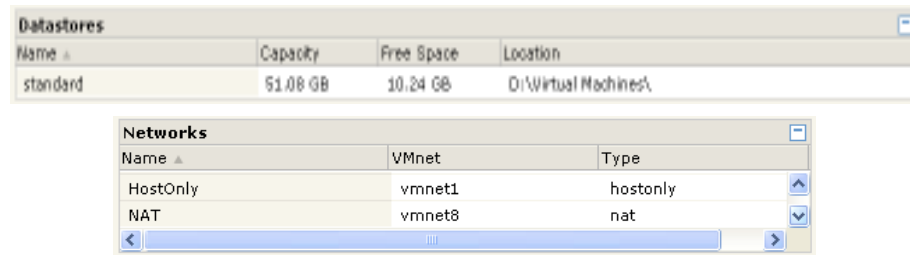
A partir de este punto y para una mejor comprensión, vamos a separar todo el proceso en varios subapartados.

Validación de la red virtual

El primer paso a realizar ha sido la comprobación de las subredes creadas por Vmware Server. En nuestro caso **VMnet1** con la subred **192.168.160.0** y **VMnet8** con **192.168.201.0**, como se puede observar en la imagen inferior. El gateway utilizado es **192.168.201.2**.

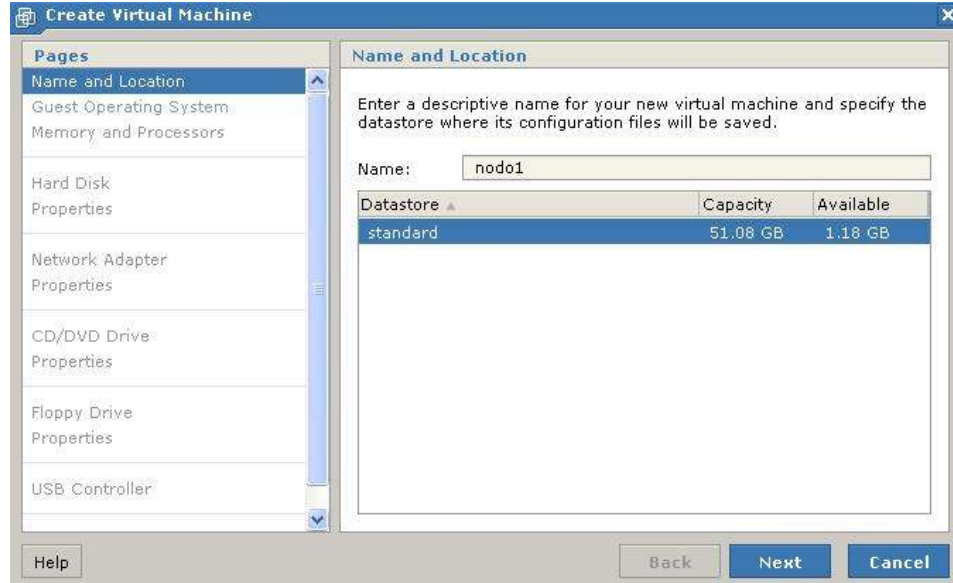


El siguiente paso es acceder a la pantalla principal de nuestro Vmware Server y comprobar que existe un datastore llamado *standard* que apunta a *D:\Virtual Machines* y que las dos redes *vmnet1* y *vmnet8* están disponibles como *HostOnly* y *NAT* respectivamente.

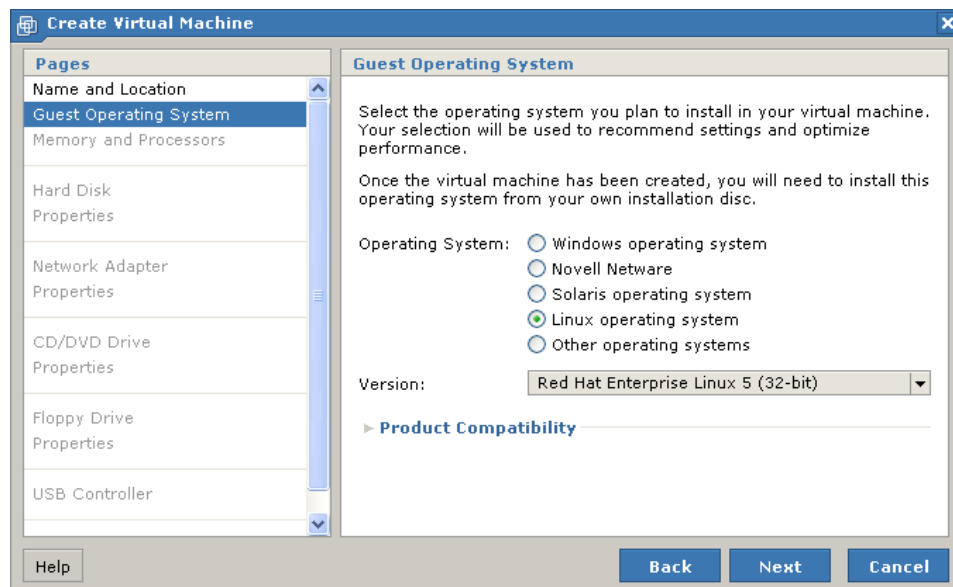


Creación de la máquina virtual

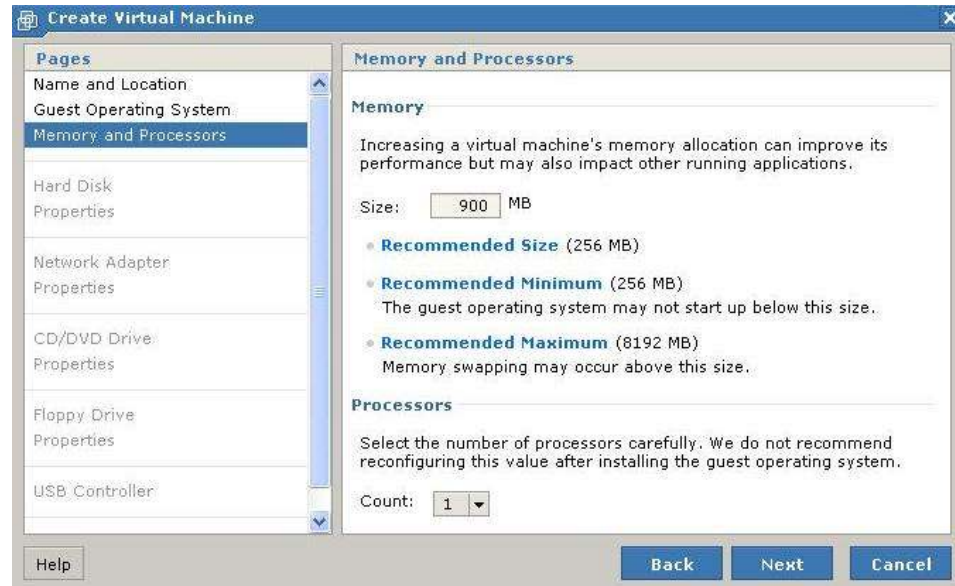
Ahora vamos a **crear la máquina virtual**. Nuestra máquina se va a llamar **nodo1**.



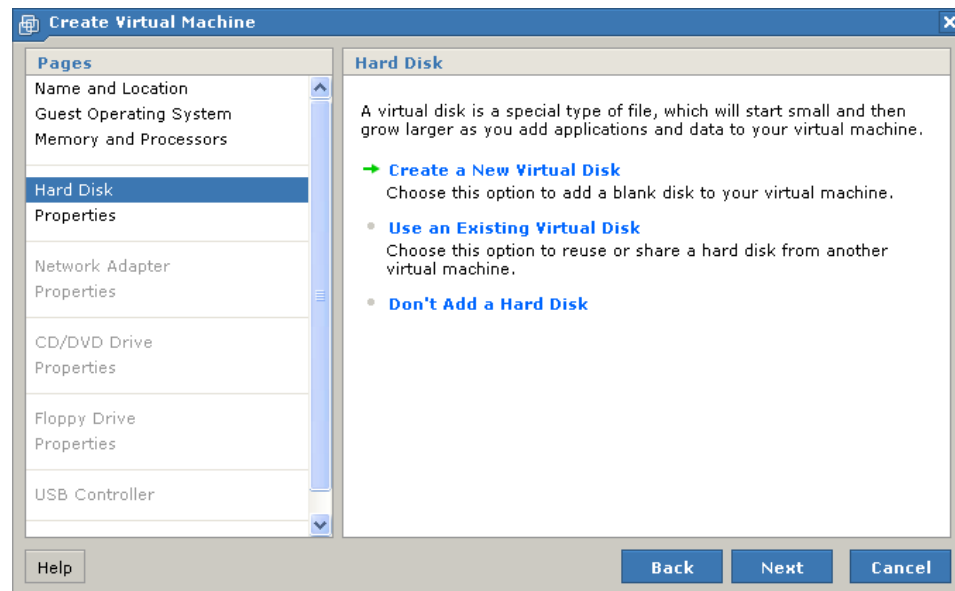
Ahora seleccionamos como sistema operativo Linux y **Red Hat Enterprise Linux 5 (32-bits)**.



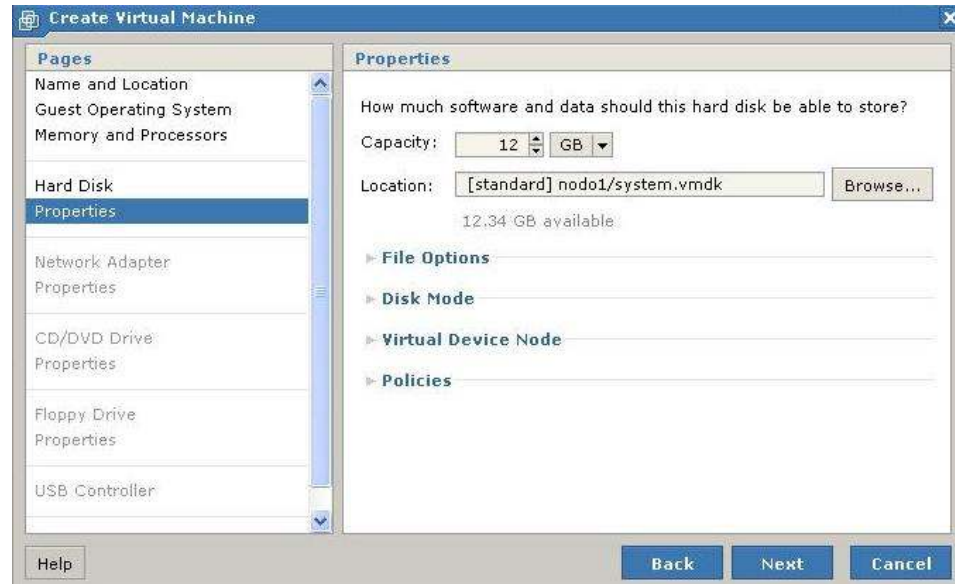
En la siguiente pantalla indicamos **900MB de memoria y 1 procesador**.



Una vez hecho esto seleccionamos la opción *Create a New Virtual Disk*.



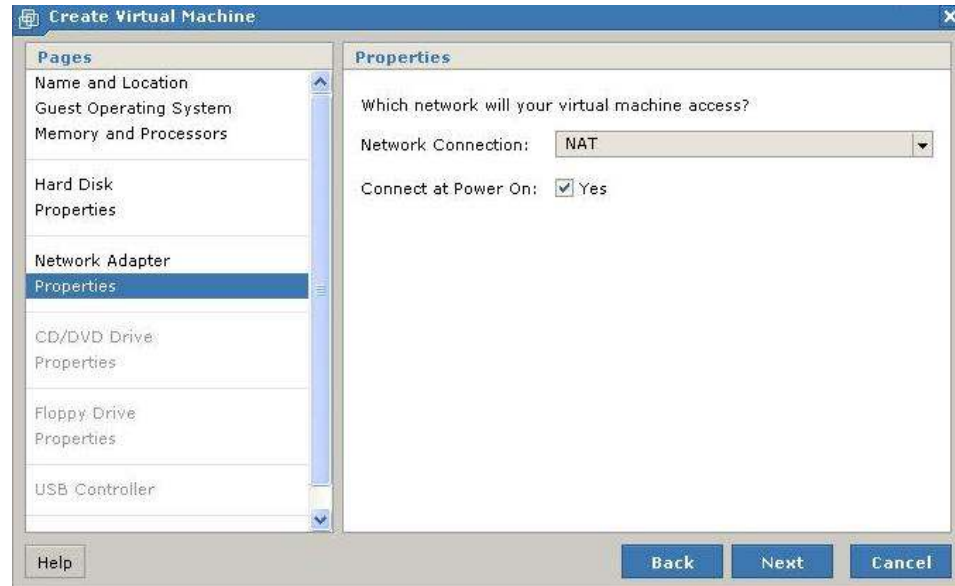
En la pantalla que nos aparece a continuación indicamos la capacidad que deseamos para este disco duro virtual, **12 GB**, y la localización donde queremos que se cree, **D:\Virtual Machines\nodo1\system.vmdk**.



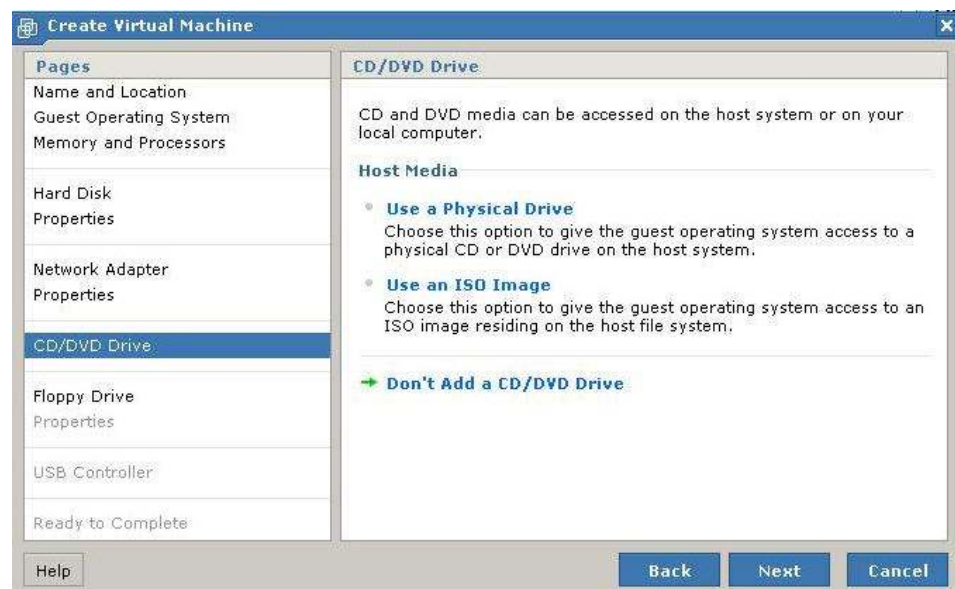
Ahora añadimos un adaptador de red.



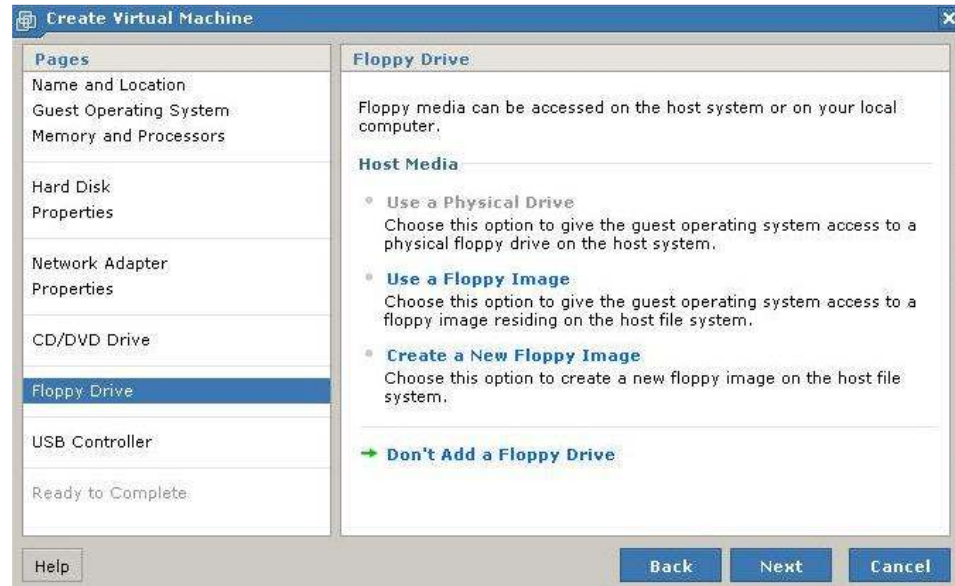
Para este primer adaptador (luego crearemos otro) utilizaremos el tipo de conexión *NAT* y elegimos que se conecte siempre al iniciar la máquina virtual.



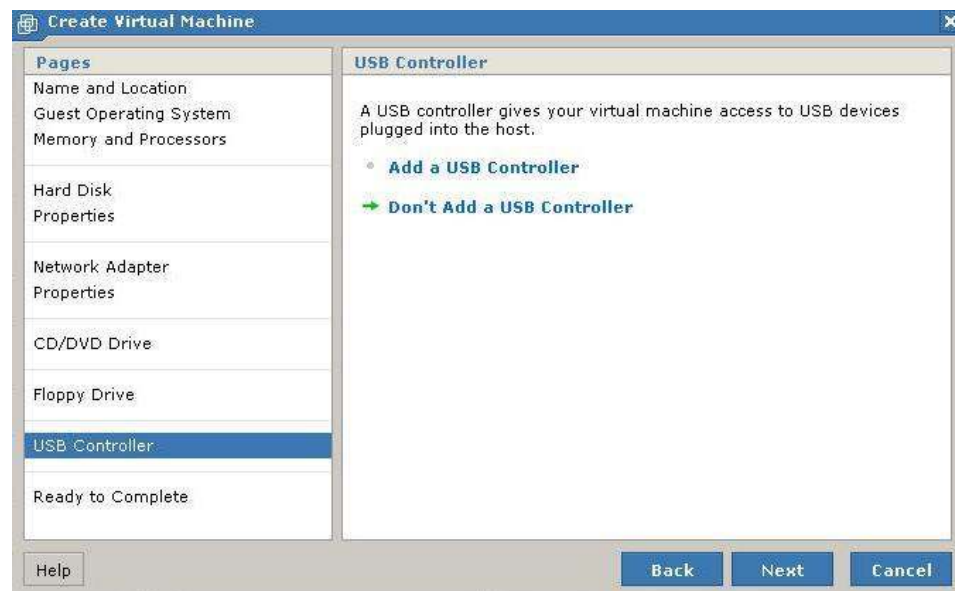
Ahora se nos pregunta sobre si deseamos añadir cdroms. Elegimos no añadir ninguno.



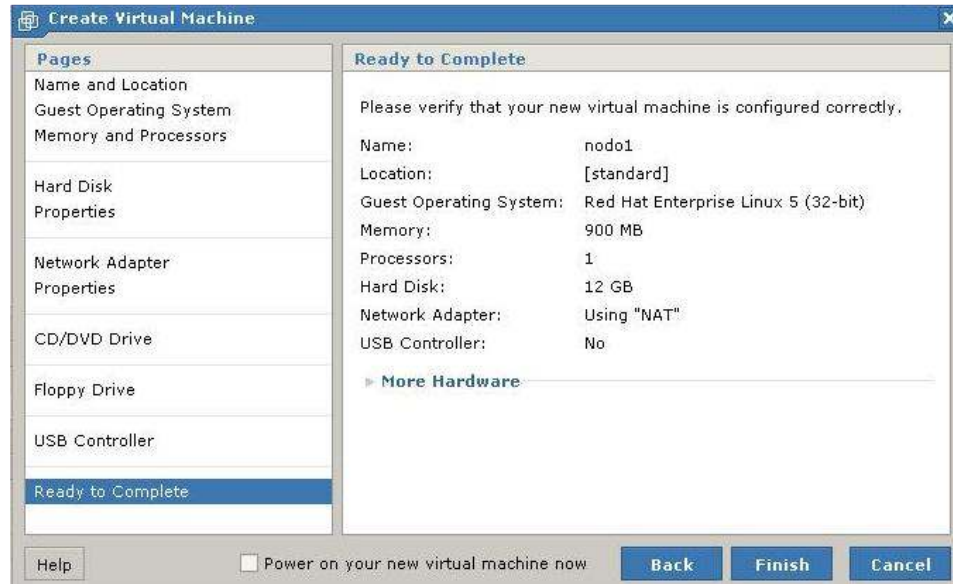
Al igual que para los cdroms, elegimos no añadir ningún floppy.



Tampoco añadimos dispositivos USB.



Por último comprobamos la configuración de nuestra máquina virtual y clicamos sobre *Finish* para terminar.



Tras estas operaciones pasamos a la configuración de 4 lectores de cdrom. Estos estarán ligados con las imágenes que tenemos del sistema operativo en *D:\Virtual Machines\iso*.

Los 4 dispositivos son los siguientes:

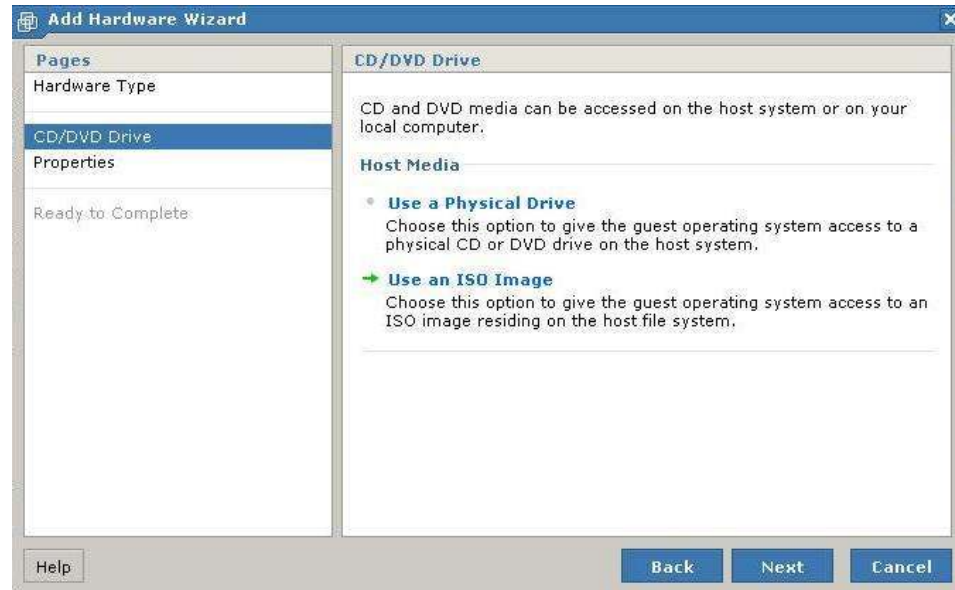
IDE 0:0 [standard] iso/Enterprise-R5-U3-Server-i386-disc1.iso
IDE 0:1 [standard] iso/Enterprise-R5-U3-Server-i386-disc2.iso
IDE 1:0 [standard] iso/Enterprise-R5-U3-Server-i386-disc3.iso
IDE 1:1 [standard] iso/Enterprise-R5-U3-Server-i386-disc4.iso

A continuación mostramos la configuración del primero:

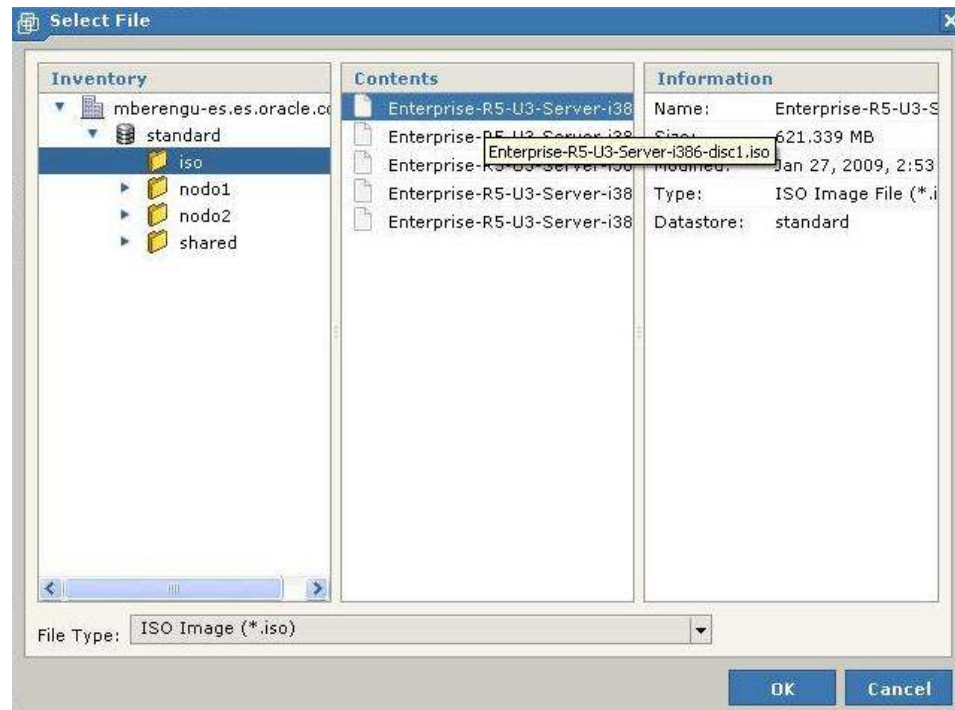
En la pantalla de administración de nuestra máquina virtual, clicamos en la opción *Add Hardware*, situada en el menú *Commands*. En la pantalla que nos aparece clicamos sobre *CD/DVD Drive*.



Seleccionamos la opción de **utilizar una imagen ISO**.



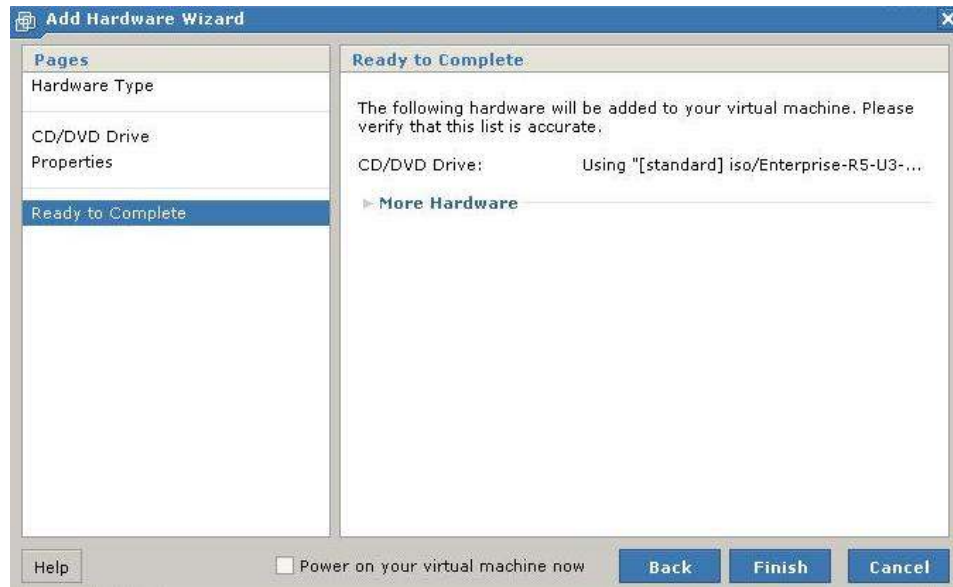
Elegimos la imagen que queremos utilizar. En este caso la **primera imagen de Enterprise Linux** situada, como ya hemos comentado, en nuestra localización de almacenamiento standard, en la carpeta *iso*.



Una vez seleccionada la imagen le indicamos que se conecte con el arranque y el dispositivo y adaptador que queremos, en este caso *IDE 0:0*.



En esta última pantalla comprobamos que la configuración es correcta y clicamos sobre *Finish* para crear el dispositivo virtual.



Estos pasos que acabamos de describir para crear el primer dispositivo los **repetimos 3 veces** más para crear los dispositivos de cdrom que hemos indicado anteriormente. Una vez realizados todos los pasos con los dispositivos la máquina virtual queda configurada como se muestra a continuación:

Hardware		
Processors		1
Memory		900 MB
Hard Disk 1 (SCSI 0:0)		12.00 GB
Network Adapter 1		NAT
CD/DVD Drive 1 (IDE 0:0)		Using file Enterprise-R5-U3-Server-i386-disc1.iso
CD/DVD Drive 2 (IDE 0:1)		Using file Enterprise-R5-U3-Server-i386-disc2.iso
CD/DVD Drive 3 (IDE 1:0)		Using file Enterprise-R5-U3-Server-i386-disc3.iso
CD/DVD Drive 4 (IDE 1:1)		Using file Enterprise-R5-U3-Server-i386-disc4.iso
SCSI Controller 0		LSI Logic

Instalación del sistema operativo

El siguiente paso, una vez creada la máquina virtual, es su inicialización y la **instalación de Enterprise Linux Red Hat 5**. Para ello, después de instalar un plug-in necesario para la consola, iniciamos la instalación arrancando la máquina virtual:

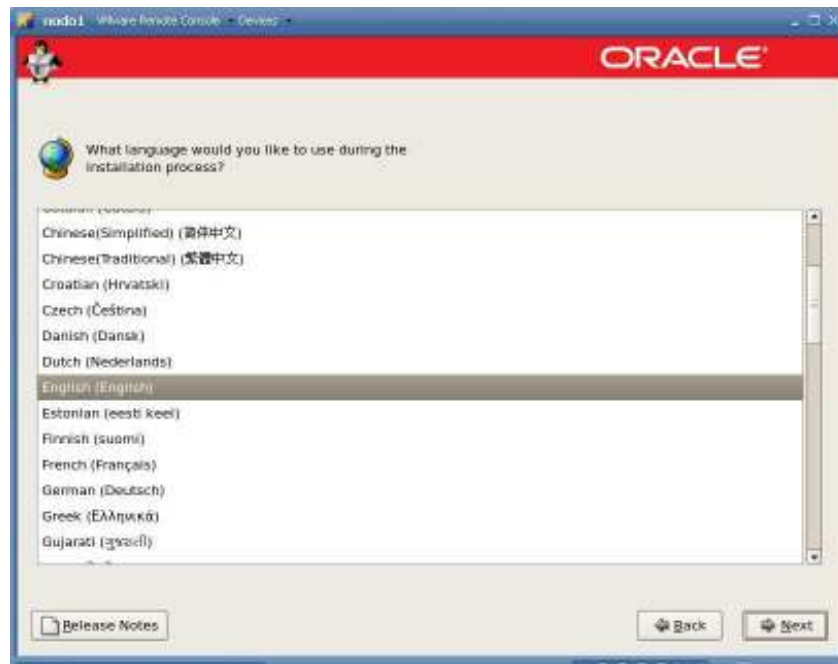
Una vez cargada la consola de la máquina virtual nos aparece la siguiente pantalla en la cual elegimos la opción de **saltar el testeo pre-instalación**. Presionamos *Skip*.



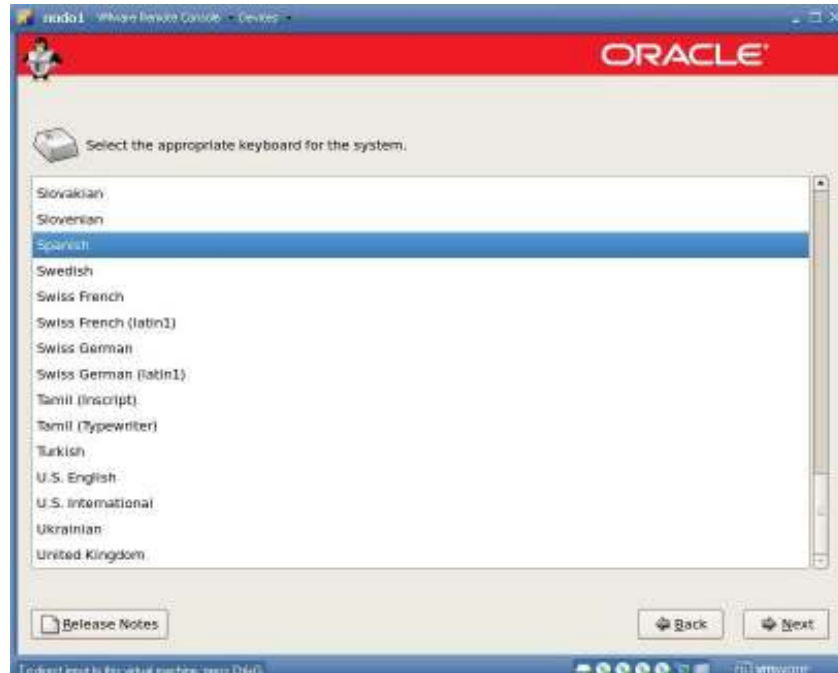
Ahora, en la primera pantalla de instalación, simplemente clicamos sobre *Next*.



Seleccionamos el **lenguaje de la instalación**, inglés.



Para el teclado elegimos la **configuración española**.



Ahora nos aparece una pantalla con una advertencia sobre la inicialización del dispositivo sda. Clicamos sobre Yes para continuar.



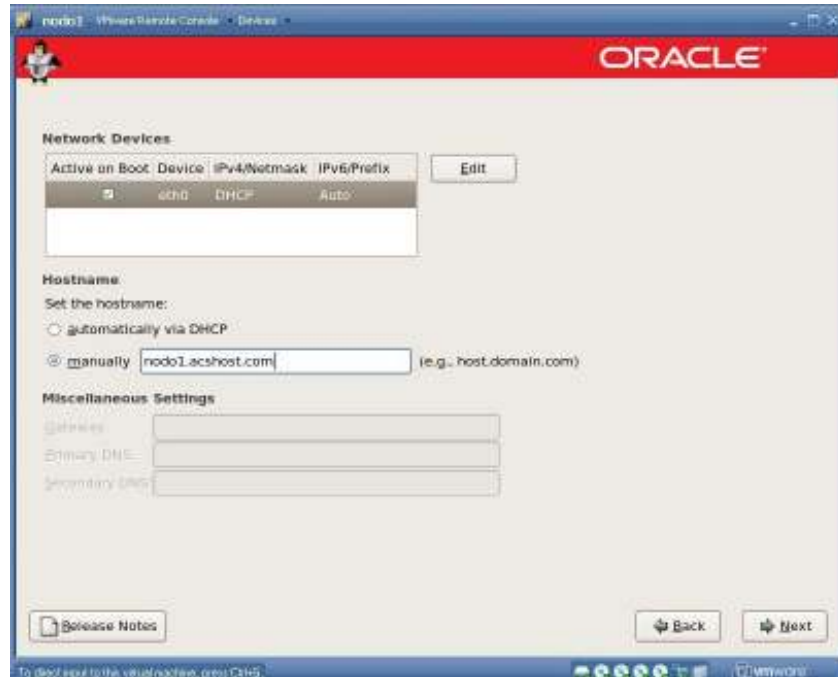
Elegimos la opción por defecto sin encriptación y pasamos a la siguiente pantalla.



En la advertencia que nos aparece a continuación clicamos en *Yes* para eliminar todas las particiones.



Ahora introducimos el nombre del host de nuestra máquina virtual, **nodo1.acshost.com**, y dejamos seleccionado la opción de DHCP.



Elegimos nuestra zona horaria, **Europe/Madrid**.



Introducimos las contraseñas para nuestro **usuario root**.



Elegimos **Customize now** para configurar manualmente las opciones de la instalación.



Elegimos los siguientes paquetes:

Desktop Environments	GNOME Desktop Environment
Applications	Editors Graphical Internet Text-based Internet
Development	Development Libraries Development Tools
Servers	Server Configuration Tools
Base System	Administration Tools Base System Tools X Window System



Una vez elegidos los paquetes llegamos a la pantalla de inicio de instalación. Clicamos sobre *Next* para comenzar.



Al iniciar la instalación nos aparece una pantalla de advertencia sobre los discos que debemos tener dispuestos. Esto ya está configurado, por lo que clicamos sobre *Continue*.



Una vez **finalizada la instalación** aparece la pantalla que mostramos a continuación. Elegimos la opción *Reboot* para reiniciar.



Tras la reiniciación de la máquina virtual nos aparece una nueva pantalla de bienvenida. Clicamos sobre *Forward*.



Ahora aceptamos la licencia y continuamos presionando *Forward*.



Deshabilitamos el firewall y clicamos *Forward*. Nos aparece un mensaje de advertencia sobre el firewall y el sistema de seguridad. Clicamos sobre *Yes*.



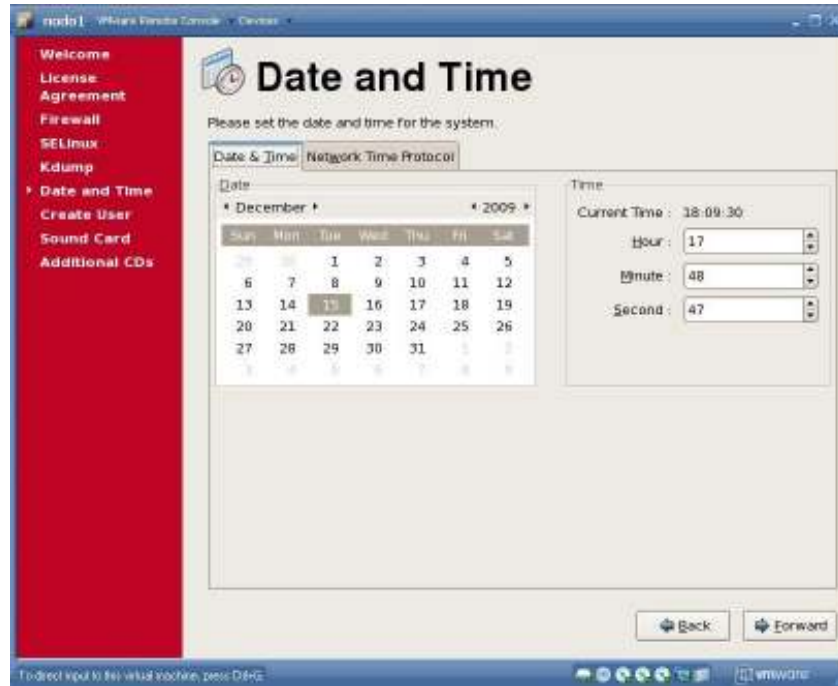
Ahora deshabilitamos el Security Enhanced Linux (SELinux) y clicamos *Forward* para continuar. Nos aparece un nuevo mensaje de advertencia sobre estas configuraciones en el cual presionamos sobre *Yes*.



Dejamos Kdump deshabilitado y presionamos *Forward*.



Marcamos la fecha y dejamos el reloj tal como está (más adelante lo sincronizaremos con el de nuestra máquina principal). Clicamos en *Forward*.



En la nueva pantalla no creamos un nuevo usuario. Nuestro usuario, además de root, será oracle pero lo haremos más adelante ya con el sistema operativo configurado totalmente. Presionamos *Forward* y *Continue* para confirmar.



Clicamos *Forward* y obviamos la configuración de la tarjeta de sonido.

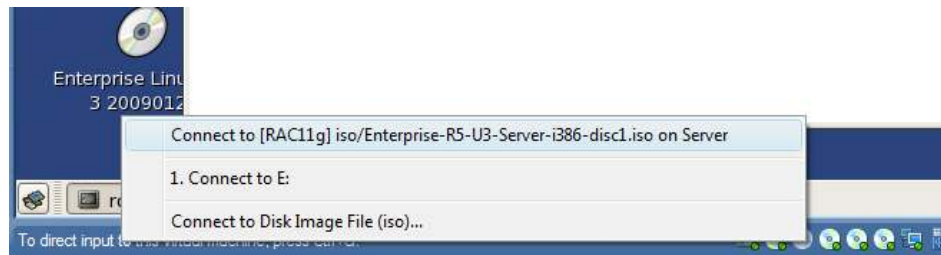


Ya se ha **finalizado la instalación**. Ahora ya se puede acceder a nuestra máquina virtual con el sistema operativo ejecutándose. Sin embargo, las labores de configuración anteriores al objetivo del proyecto todavía no han concluido.

Configuración del sistema operativo

Una vez instalado e iniciado ya nuestro sistema operativo, arrancamos con nuestro usuario `root` y **montamos los discos virtuales** que hemos creado anteriormente. En estos discos se encuentran rpms que luego necesitaremos para la puesta a punto del entorno antes de las instalaciones de los productos.

Antes de montar los discos debemos clicar sobre un icono que aparece debajo de la consola de la máquina virtual y conectar el dispositivo a la imagen que le habíamos indicado en pasos anteriores, tal como se muestra a continuación:



Tras esto, para montar los discos ejecutamos:

```
[root@nodo1 mnt]# mkdir cdrom
[root@nodo1 mnt]# mkdir cdrom2
[root@nodo1 mnt]# mkdir cdrom3
[root@nodo1 mnt]# mkdir cdrom4
[root@nodo1 mnt]# mkdir cdrom5
```

Ahora, editamos el archivo **fstab** y añadimos las siguientes líneas:

```
[root@nodo1 ~]# gedit /etc/fstab
```

```
/dev/cdrom-hda /mnt/cdrom iso9660 defaults 0 0  
/dev/cdrom-hdb /mnt/cdrom2 iso9660 defaults 0 0  
/dev/cdrom-hdc /mnt/cdrom3 iso9660 defaults 0 0  
/dev/cdrom-hdd /mnt/cdrom4 iso9660 defaults 0 0  
/dev/cdrom-sr0 /mnt/cdrom5 iso9660 defaults 0 0
```

Por último, montamos las cinco unidades:

```
[root@nodol mnt]# mount cdrom  
mount: block device /dev/cdrom-hda is write-  
protected, mounting read-only  
[root@nodol mnt]# mount cdrom2  
mount: block device /dev/cdrom-hdb is write-  
protected, mounting read-only  
[root@nodol mnt]# mount cdrom3  
mount: block device /dev/cdrom-hdc is write-  
protected, mounting read-only  
[root@nodol mnt]# mount cdrom4  
mount: block device /dev/cdrom-hdc is write-  
protected, mounting read-only  
[root@nodol mnt]# mount cdrom5  
mount: block device /dev/cdrom-hdc is write-  
protected, mounting read-only
```

Para conseguir una mayor facilidad y comodidad en el manejo entre nuestra máquina virtual y nuestro sistema operativo padre vamos a instalar las **VMware Tools**. Estas herramientas permiten pasar el control entre la VMware y nuestro sistema normal de forma sencilla y cómoda (sin tener que clicar en la ventana para que se fije el control) y copiar y pegar texto entre ambos, entre otras funcionalidades.

Antes de instalar debemos desmontar la unidad `cdrom`:

```
[root@nodol mnt]# eject /mnt/cdrom
```

Para instalar ya las VMware Tools clicamos en **Install VMware Tools** dentro del panel **Status** en la consola de VMware. Tras esto, nos aparece una pantalla pidiéndonos confirmación para su instalación. Clicamos en **Install**:



Ahora remontamos la primera unidad de `cdrom`:

```
[root@nodol mnt]# mount /mnt/cdrom  
mount: block device /dev/cdrom-hda is write-protected, mounting  
read-only
```

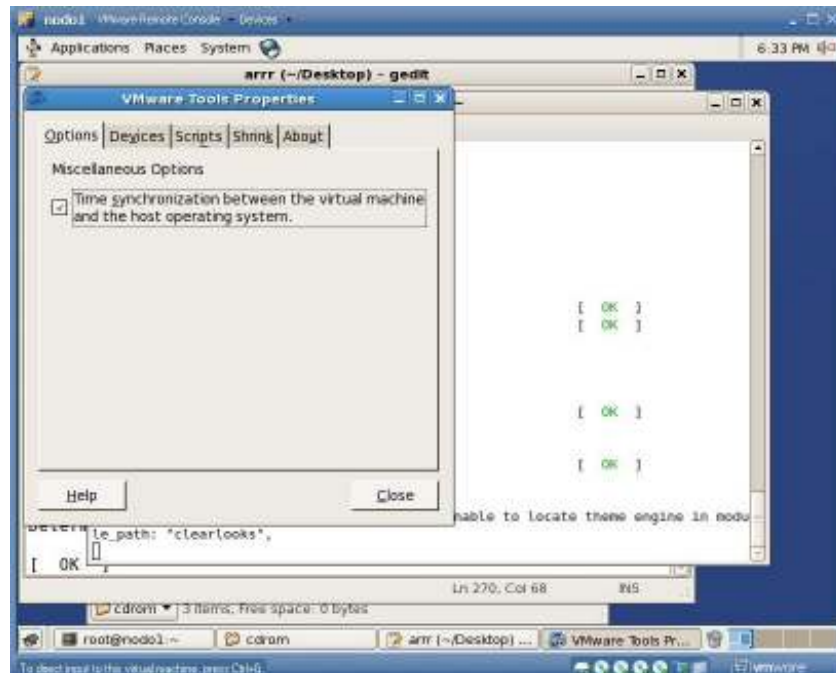
E instalamos el rpm con las VMware Tools:

```
[root@nodol mnt]# rpm -ivh cdrom/VMwareTools-7.7.5-  
156745.i386.rpm  
Preparing...  
##### [100%]  
1:VMwareTools  
##### [100%]
```

Por último configuramos las Vmware Tools mediante:

```
[root@nodol cdrom]# vmware-config-tools.pl
```

Esta herramienta realiza varias configuraciones y nos permite la sincronización de reloj entre nuestro sistema padre y la máquina virtual, cosa que mostramos a continuación:



Después de acabar la instalación de las Vmware Tools cerramos la sesión y reiniciamos la máquina. Tras la reiniciación las Vmware Tools estarán ya cargadas y nos será más fácil el trabajo con la máquina virtual.

Instalación de la base de datos 10.2.0.4

En este paso se podría mostrar el proceso de instalación y configuración de la base de datos 10.2.0.4, pero **se ha obviado ya que la instalación es prácticamente igual a la 11.1.0.6** y ésta es mostrada en un apartado posterior.

Con el propósito de confirmar posteriormente que ciertos datos se han migrado correctamente creamos una serie de datos fáciles de comprobar.

Conectados como sysdba **creamos un nuevo usuario Miguel** con tablespace por defecto *users*:

```
SQL> create user miguel identified by oraMiguel  
      default tablespace users;
```

Comprobamos que nuestro usuario, su tablespace y perfil están creados mediante la siguiente consulta:

```
SQL> select username, default_tablespace, profile  
      from dba_users where username = 'MIGUEL';
```

```
USERNAME                                DEFAULT_TABLESPACE  
-----  
PROFILE
```

```
-----  
MIGUEL                                USERS  
DEFAULT
```

Le añadimos el rol *dba* para que tenga los privilegios de administrador y pueda crear, consultar, actualizar y eliminar tablas, vistas, sesiones,...:

```
SQL> GRANT dba TO miguel;  
User altered.
```

Nos conectamos ahora como usuario miguel:

```
SQL> connect miguel  
Enter password:  
Connected.  
SQL> show user  
USER is "MIGUEL"
```

Creamos una nueva tabla que pertenezca a este usuario. La tabla se llama albums y la creamos con nomAlbum como clave primaria:

```
create table albums (nomAlbum varchar2(100) not null,  
grupo varchar2(100) null, fLanzamiento number null,  
nCanciones number null, constraint pk_albums primary  
key (nomAlbum));
```

Ahora **introducimos una fila** en esta nueva tabla de esta forma:

```
SQL> insert into albums (nomAlbum, grupo,  
fLanzamiento, nCanciones) VALUES ('Night fall in  
Middle Earth', 'Blind Guardian',1998,22);  
1 row created.
```

Consultamos la tabla y fila creadas:

```
SQL> select * from albums;  
NOMALBUM  
-----  
GRUPO  
-----  
FLANZAMIENTO NCANCIONES  
-----  
Night fall in Middle Earth  
Blind Guardian  
1998 22
```

Por último, para que quede constancia de la **versión de la base de datos y características actuales** ejecutamos:

```
SQL> select * from v$instance;  
INSTANCE_NUMBER INSTANCE_NAME  
-----  
HOST_NAME  
-----  
VERSION          STARTUP_T STATUS      PAR  
-----  
THREAD# ARCHIVE LOG_SWITCH_WAIT  
-----  
LOGINS          SHU DATABASE_STATUS  INSTANCE_ROLE
```

```
-----  
ACTIVE_ST BLO  
-----  
1 DBase10  
nodo1.acshost.com  
10.2.0.4.0 16-SEP-09 OPEN YES  
1 STOPPED  
ALLOWED NO ACTIVE PRIMARY_INSTANCE  
NORMAL NO
```

Y aquí damos por finalizado este apartado. En el siguiente apartado veremos todas las configuraciones necesarias para preparar nuestro entorno de cara a instalar todos los productos necesarios y realizar la migración y el paso a RAC.

Premigración

El primer paso a la hora de realizar una migración es la comprobación de que se cumplen los requisitos mínimos para realizarla. Esto pasa por hacer un estudio de la totalidad del sistema a nivel de hardware, software, red, sistemas de almacenamiento,... y comprobar su compatibilidad y certificación con el software que se va a instalar.

La premigración consta de varios apartados en los que iremos mostrando la realización de las pruebas sobre todos los elementos del sistema para la comprobación de su compatibilidad con los requisitos mínimos del nuevo software y sus certificaciones.

Así pues, la Premigración se va a dividir en:

- Preparación para Clusterware.
- Preparación para RAC.
- Preparación para la Base de datos y ASM.
- Preparación del almacenamiento.

Preparación para Clusterware

En este apartado trataremos todos los aspectos que se deben realizar antes de la instalación del clusterware de modo que en el apartado **Ejecución de la migración** ya se hayan realizado y podamos dedicar en exclusiva a la configuración del sistema con el nuevo software y características.

En algunos casos cabe recordar que ya habrán sido configurados muchos puntos ya que el sistema ya contiene una base de datos 10.2.0.4 funcionando y muchas de sus configuraciones son válidas para los demás productos a instalar.

En el caso del almacenamiento requerido para Clusterware se hace una explicación en profundidad en el apartado **Preparación del almacenamiento**. Esto es así para que sea más fácil su comprensión y para mostrar las configuraciones necesarias para Clusterware, ASM y RAC al mismo tiempo.

Grupos y usuarios necesarios para Clusterware

Oracle Clusterware necesita un grupo y usuario obligatoriamente: el grupo **Oracle Inventory** y el usuario propietario del software del Clusterware, típicamente **oracle**. Se podría crear un nuevo usuario propietario para Clusterware pero, como ya tenemos el usuario **oracle** de la base de datos, utilizaremos el mismo para todo el software Oracle.

En nuestro caso, como vamos a realizar una migración sobre un sistema Oracle, la mayoría de grupos y el usuario oracle ya están configurados en nuestro sistema ya que son necesarios también para la configuración y funcionamiento del sistema que tenemos actualmente. Igualmente, mostraremos como se crean.

En nuestro nodo1 el grupo Oracle Inventory se llama **oinstall** y el grupo propietario **oracle**. Mostramos, aunque ya está creado, la creación del grupo **oinstall**:

```
[root@nodo1 ~]# groupadd oinstall
```

Para comprobar el usuario **oracle** ejecutaremos:

```
[root@nodo1 ~]# id oracle
```



```
uid=500(oracle) gid=500(oinstall)  
groups=500(oinstall),501(dba),502(oper),  
503(asmadmin)
```

demostrando su existencia y su pertenencia al **oinstall**. En este nodo, como se puede comprobar, también existen los grupos **dba** y **oper**, que son también referentes a la base de datos, con otros privilegios. Estos grupos también fueron creados anteriormente para la instalación de la base de datos 10.2.0.4 de la siguiente forma:

```
[root@nod01 ~]# groupadd dba  
[root@nod01 ~]# groupadd oper  
  
[root@nod01 ~]# groupadd asmadmin  
[root@nod01 ~]#  
[root@nod01 ~]# useradd -u 500 -g oinstall -G  
dba,oper,asmadmin oracle  
[root@nod01 ~]# passwd oracle  
Changing password for user oracle.  
New UNIX password:  
BAD PASSWORD: it is based on a dictionary word  
Retype new UNIX password:  
passwd: all authentication tokens updated  
successfully.
```

El directorio **Oracle Inventory** que se ha nombrado antes es el directorio donde se comparte el inventario de todo el software de Oracle instalado en el sistema. Ya está configurado en el `nod01`. Su localización, ya configurada anteriormente, la consultamos mediante:

```
[root@nod01 ~]# more /etc/oraInst.loc  
inventory_loc=/u01/app/oraInventory  
inst_group=oinstall
```

`oraInst.loc` es el archivo que identifica el nombre del grupo Oracle Inventory y el path del Oracle Inventory directory.

El último paso en este apartado es la creación del path para `$CRS_HOME`, es decir, el **home para Clusterware**. El nuestro será `/u01/crs/oracle/product/11.1.0/crs` y para crearlo ejecutamos:

```
[root@nod01 ~]# mkdir -p  
/u01/crs/oracle/product/11.1.0/crs
```

Requisitos hardware para Clusterware

Lo primero a tener en cuenta es que todos los servidores del Clusterware deben utilizar la misma **arquitectura de chip**. En nuestro caso es evidente que no habrá problema al un sólo nodo. La arquitectura es de 32-bits.

En cuanto a la memoria **RAM** y el **Swap**, se requiere que teniendo 2GB de RAM, el tamaño del swap sea equivalente al de RAM. En nuestro caso, debido a nuestra configuración con máquinas virtuales no se cumple con las exigencias totalmente, sin embargo, sí lo suficiente como para poder desarrollar el proyecto:

```
[oracle@nod01 ~]$ grep MemTotal /proc/meminfo  
MemTotal:          909260 kB  
[oracle@nod01 ~]$ grep SwapTotal /proc/meminfo  
SwapTotal:         1835000 kB
```

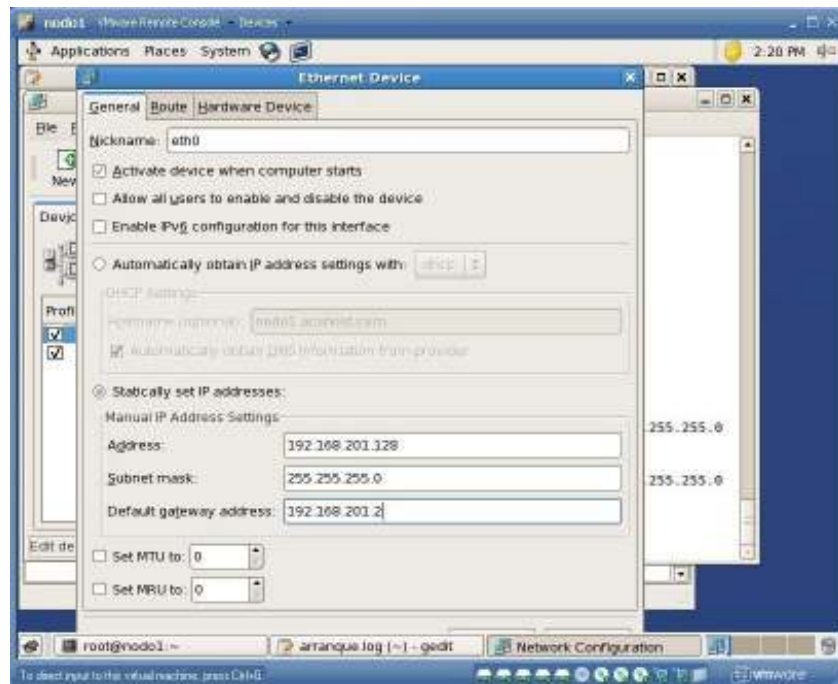
Igualmente se debe tener al menos 400MB de espacio libre en `/tmp`, requisito ampliamente cumplido.

Requisitos hardware para la red de Clusterware

Un cluster está formado por uno o más servidores, en nuestro caso lo estará por uno. El cluster necesita de una segunda red llamada red de interconexión o **interconnect network**. Esto hace necesario que los servidores tengan al menos dos tarjetas de red: una para la red privada y otra para la pública. Ambas deben soportar TCP/IP y UDP para el funcionamiento deseado. La red de interconexión es una red privada que utiliza un switch (o varios) que sólo los nodos del cluster pueden acceder.

Necesitaremos configurar:

- Una red pública, es decir, una LAN. En nuestro entorno el nodo1 tendrá la IP 192.168.201.128. Esto lo podemos hacer mediante la tarjeta de red que ya tenemos configurada en la máquina. El resultado de esta configuración lo mostramos en el punto siguiente.
- Un interfaz de red redundante para la red pública. La interfaz de red será la `eth0`. Al igual que lo anterior, con la tarjeta e interfaz que tenemos configurada es suficiente. Mostramos cómo hemos configurado los dos puntos a continuación:

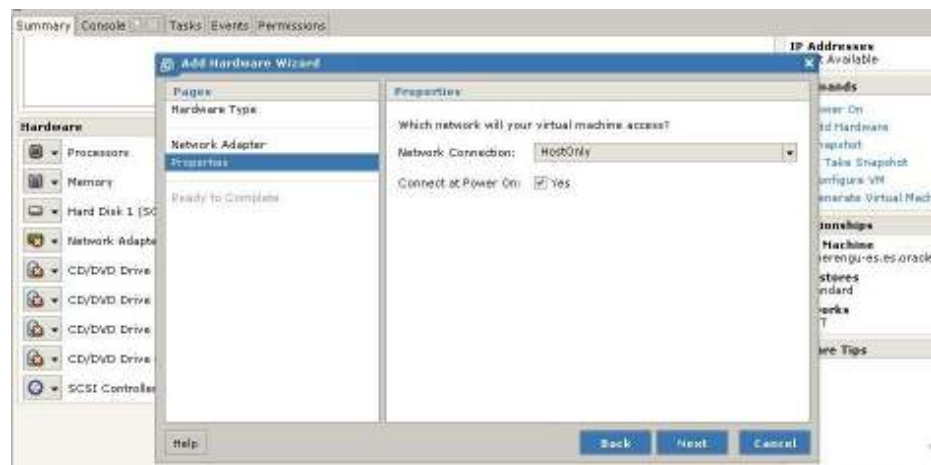


- Una red de interconexión. En este caso el nodo1 tendrá la IP 192.168.160.128. Para esta segunda red necesitamos una segunda tarjeta de red. Así pues, debemos configurar nuestra máquina para poder hacerlo.

En la consola de administración de la máquina virtual clicamos sobre *Add Hardware* y en la pantalla que nos aparece, sobre Network Adapter como se muestra en la siguiente captura:

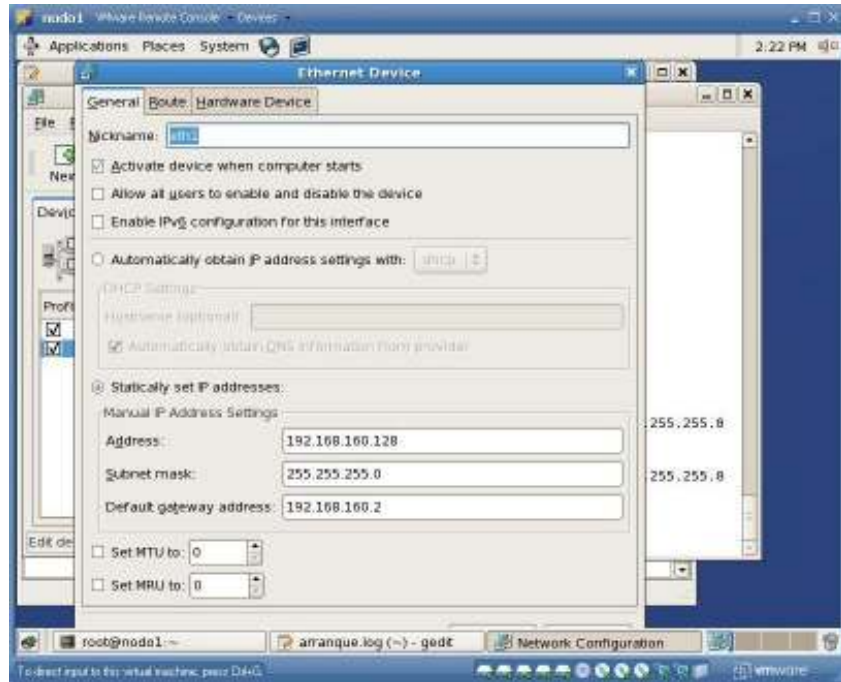


Ahora, la configuramos como HostOnly y con arranque automático con la máquina virtual:



Así ya tenemos la tarjeta que necesitamos y podemos configurar IP. Esto lo mostramos en el siguiente punto.

- Un interfaz de red redundante para la red privada de interconexión. En este caso la interfaz es la **eth1**. Ahora, tras la configuración que hemos mostrado y el reinicio de la máquina realizamos lo siguiente:



- Una IP Virtual para cada nodo que esté en la misma subred que la pública. Para el nodo1 será 192.168.201.138. Esto lo dejamos configurado en el archivo /etc/hosts quedando de la siguiente manera:

```
192.168.201.128 nod01 nod01.acshost.com
192.168.201.138 nod01-vip nod01-vip.acshost.com
192.168.160.128 nod01-priv nod01-priv.acshost.com
```

- Editamos el archivo /etc/resolv.conf para indicar nuestro host y nombre de servidor:

```
vi /etc/resolv.conf
; generated by /sbin/dhclient-script
search acshost.com
nameserver 192.168.201.2
```

Por último reiniciamos el servicio de red y comprobamos que nuestras configuraciones han tenido efecto:

```
[root@nod01 ~]# service network restart
Shutting down interface eth0:      [ OK ]
Shutting down interface eth1:      [ OK ]
Shutting down loopback interface:  [ OK ]
Bringing up loopback interface:    [ OK ]
Bringing up interface eth0:        [ OK ]
Bringing up interface eth1:        [ OK ]

[root@nod01 ~]# ifconfig | grep net
eth0 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:60
      inet addr:192.168.201.128 Bcast:192.168.201.255
      Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:fec0:b160/64
      Scope:Link
eth1 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:6A
      inet addr:192.168.160.128 Bcast:192.168.160.255
      Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:fec0:b16a/64
      Scope:Link
      inet addr:127.0.0.1 Mask:255.0.0.0
```

```
inet6 addr: ::1/128 Scope:Host
```

Requisitos software para Clusterware

En este apartado vamos a realizar chequeos y, en su caso, instalaciones de todos los paquetes y componentes software que es necesario tener listos antes de iniciar la instalación de Clusterware en nuestras máquinas.

Como se ha podido comprobar ya, la versión de Linux que utilizamos, **Red Hat 5, está certificada**. Por lo tanto, podemos pasar a la comprobación del resto del software.

Lo primero que comprobamos es la versión del **kernel**. Para el Linux que utilizamos, la versión requerida de kernel es la 2.6.9. Requisito que comprobamos que se cumple mediante:

```
uname -r  
2.6.18-8.E15.
```

A continuación mostramos la lista de todos los **paquetes** que deben estar instalados:

```
binutils-2.17.50.0.6-2.e15  
compat-libstdc++-33-3.2.3-61  
elfutils-libelf-0.125  
elfutils-libelf-devel-0.125  
glibc-2.5-12  
glibc-common-2.5-12  
glibc-devel-2.5-12  
glibc-headers-2.3.4-2  
gcc-4.1.1-52  
gcc-c++-4.1.1-52  
libaio-0.3.106  
libaio-devel-0.3.106  
libgcc-4.1.1-52  
libstdc++-4.1.1  
libstdc++-devel-4.1.1-52.e15  
make-3.81-1.1  
sysstat-7.0.0  
unixODBC-2.2.11  
unixODBC-devel-2.2.11
```

Para su comprobación utilizamos el comando:

```
rpm -q nombre_paquete
```

El resultado de esta comprobación debe ser la versión requerida o una superior. De todas formas, al final de este apartado **mostraremos la instalación de todos los paquetes necesarios para todo el software que vamos a instalar**.

Para la instalación de la base de datos 10.2.0.4 ya hemos tenido que instalar la mayoría de estos paquetes en estas versiones o anteriores. De todas formas, en los casos en que falla la versión recurrimos a los cdroms que hemos configurado en la instalación, los cuales contienen todos los rpms que necesitamos.

Ahora veremos la configuración que deben tener los **parámetros del kernel**:

```
semmsl 250  
semnms 32000  
semopm 100  
semmni 128  
shmmax 2147483648  
shmmni 4096  
shmall 2097152  
file-max 65536  
ip_local_port_range mínimo1024 - máximo 65000  
rmem_default 262144  
rmem_max 4194304  
wmem_default 262144
```

```
wmem_max 4194304  
aio-max-nr máximo 1048576
```

Para comprobar los parámetros lo podemos hacer mediante la ejecución del siguiente comando, cambiando parámetro por el nombre del que queremos comprobar:

```
/sbin/sysctl -a | grep parámetro
```

Sin embargo, lo que hacemos es **editar el archivo `sysctl.conf`** mediante `gedit` añadiendo lo siguiente:

```
kernel.shmmni = 4096  
# semaphores: semmsl, semmns, semopm, semmni  
kernel.sem = 250 32000 100 128  
net.ipv4.ip_local_port_range = 1024 65000  
net.core.rmem_default=4194304  
net.core.rmem_max=4194304  
net.core.wmem_default=262144  
net.core.wmem_max=262144
```

Finalmente los parámetros quedan configurados de la siguiente manera:

```
kernel.shmall = 2097152  
kernel.shmmax = 2147483648  
kernel.shmmni = 4096  
kernel.sem = 250 32000 100 128  
fs.file-max = 6553600  
fs.aio-max-nr=1048576  
net.ipv4.ip_local_port_range = 1024 65000  
net.core.rmem_default = 4194304  
net.core.rmem_max = 4194304  
net.core.wmem_default = 262144  
net.core.wmem_max = s262144
```

Configuración de SSH en los nodos

Antes de instalar y utilizar Clusterware se debe configurar SSH en todos los nodos del cluster con el usuario con el que está planificado que se haga la ejecución, en nuestro caso **oracle**.

OUI (Oracle Universal Instaler) utiliza SSH para ejecutar comandos durante la instalación y copiar archivos, con lo que se hace indispensable su instalación y configuración previa.

Primero comprobaremos que el nodo tiene ssh instalado mediante:

```
[oracle@nod01 ~]$ cd .ssh/  
[oracle@nod01 ~]$ pgrep sshd
```

Lo que nos devuelve 2360, es decir, el ID del proceso de ssh. Luego comprobamos que el usuario que va a ejecutar la instalación, `oracle`, es el propietario de `.ssh` y puede escribirlo con:

```
[oracle@nod01 ~]$ ls -al
```

devolviéndonos la siguiente línea confirmando que está todo correcto:

```
drwx----- 2 oracle oinstall 4096 Apr 25 2005  
.ssh
```

El último paso es la **configuración de la equivalencia de usuario**:

```
[root@nod01 ~]# su - oracle  
[oracle@collabn1 ~]$ ssh localhost
```

```
The authenticity of host 'localhost (127.0.0.1)'
can't be established. RSA key fingerprint is
3f:65:e6:77:af:11:84:56:a4:b9:0f:8b:41:b4:2f:8a.
Are you sure you want to continue connecting
(yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the
list of known hosts. oracle@localhost's password:

[oracle@nodol ~]$ ssh-keygen -t rsa Generating
public/private rsa key pair. Enter file in which to
save the key (/home/oracle/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/oracle/.ssh/id_rsa.
Your public key has been saved in
/home/oracle/.ssh/id_rsa.pub.
The key fingerprint is:
0a:33:74:d4:7a:c0:c6:d0:56:33:10:4e:71:63:4c:e8
oracle@collabn1.ioug.org

[oracle@nodol ~]$ cat /home/oracle/.ssh/id_rsa.pub >>
/home/oracle/.ssh/authorized_keys
```

Configuración del entorno de oracle y root para Clusterware

Ahora vamos a configurar los **límites del shell para el usuario oracle** para incrementar el rendimiento del software en nuestro sistema. Estos límites son configurados en el archivo `/etc/security/limits.conf` y quedan como mostramos a continuación:

```
oracle soft nproc 2047
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
```

Mediante la modificación de `nofile` conseguimos que no se abra una cifra de descriptores de fichero superior a 65536 al igual que restringimos a un máximo de 16384 el número de procesos disponibles para un sólo usuario en `nproc`.

En el archivo `/etc/pam.d/login` debemos hacer referencia a los límites mediante `pam_limits.so` para lo cual, añadimos las siguientes líneas:

```
session required /lib/security/pam_limits.so
session required pam_limits.so
```

Editamos el archivo `oratab` con la información referente a nuestro clusterware:

```
[root@nodol ~]# gedit /etc/oratab
crs:/u01/crs/oracle/product/11.1.0/crs:N
```

Cambiamos los permisos y propietario del archivo:

```
[root@nodol ~]# chown oracle:dba /etc/oratab
```

Editamos el archivo de **root** `.bashrc` con:

```
[root@nodol ~]# gedit /root/.bashrc
ORAENV_ASK=NO
ORACLE_SID=crs
. oraenv >/dev/null
unset ORAENV_ASK
```

Por último volvemos al usuario `oracle` y editamos su archivo `.bash_profile` de la siguiente manera:

```
[root@nodol ~]# su - oracle
[oracle@nodol ~]$ cd /mnt
```

```
[oracle@nodol ~]$ cat */oracle-profile
>>/home/oracle/.bash_profile

export ORACLE_BASE=/u01/app/oracle
export ADMIN_TOP=$ORACLE_BASE/admin
export ORACLE_SID=RAC1
ORAENV_ASK=NO
grep -q "^$ORACLE_SID" /etc/oratab && . oraenv
unset ORAENV_ASK
if [ $USER = "oracle" ]; then
    if [ $SHELL = "/bin/ksh" ]; then
        ulimit -p 16384
        ulimit -n 65536
    else
        ulimit -u 16384 -n 65536
    fi
fi
export PATH=$PATH:/usr/sbin
alias s=sqlplus
alias ss="sqlplus '/ as sysdba'"
alias cdoh='cd $ORACLE_HOME'
alias cda='cd $ADMIN_TOP; cd ./$ORACLE_SID
2>/dev/null||cd ./${ORACLE_SID/%?}'
alias cdd='cd $ORACLE_BASE/diag'
alias oenv='echo "SIDs here are: $(egrep -v "(^#|^$)"
/etc/oratab|awk -F:
\''{printf$1 " }'\''"; . oraenv'
echo "Set environment by typing 'oenv' - default is
instance $ORACLE_SID."
PS1='\h:$PWD[$ORACLE_SID]\$ '
```

Preparación para la Base de datos y ASM

En este apartado trataremos, al igual que en el anterior, todas las tareas que se deben realizar antes de la migración de la base de datos de modo que en el apartado **Ejecución de la migración** ya se hayan realizado y así, podernos dedicar en exclusiva a la configuración del sistema con el nuevo software.

Además de este apartado de preparación se debe completar las tareas con el siguiente **Preupgrade de la base de datos** para asegurar que la migración se hace con la mayor seguridad.

Para configurar ASM se necesita configurar particiones y hacer diferentes tareas. Como también se necesita hacer tareas de este tipo para la configuración del almacenamiento de Clusterware, al igual que hemos comentado antes, se explica en detalle todo lo referente a **almacenamiento en un punto posterior**.

Requisitos de hardware para 11g

Como pasa con todos los productos que vamos a instalar, la base de datos requiere una serie de condiciones en cuanto al hardware.

La base de datos Oracle 11g necesita un mínimo de un 1GB de **RAM**. Viendo la guía de configuración de entornos que hemos hecho anteriormente comprobamos que ambos nodos tienen 2GB. Al tener 2GB de RAM, Oracle recomienda que el tamaño de **SWAP** sea igual que el de RAM. Como también podemos comprobar en la guía, se cumple las exigencias.

Para comprobar ambas cosas lo podemos hacer igualmente con:

```
grep MemTotal /proc/meminfo
```



```
grep SwapTotal /proc/meminfo
```

En este caso, a diferencia de Clusterware, sólo es necesario entre 150 y 200MB de espacio en el directorio **/tmp**, cosa que, al igual que con Clusterware, se supera sin problemas.

La arquitectura del sistema de nuestro nodo es i686, cosa que podemos comprobar mediante:

```
[oracle@nodo1 ~]$ uname -m  
i686
```

Con esta arquitectura el **espacio de disco** necesario para la instalación del software de la base de datos es de 3.47GB, lo cual no es ningún problema en nuestras máquinas.

Requisitos de software para 11g

Como también hemos podido comprobar en la matriz de certificación para base de datos, nuestro **Sistema Operativo**, Red Hat 5, está certificado y no hay problema para la instalación.

La **versión del kernel** que necesita la base de datos es la 2.6.18 y comprobamos que la instalada es 2.6.18-8.el5:

```
[oracle@nodo1 ~]$ uname -r  
2.6.18-128.el5
```

A continuación comprobamos los paquetes requeridos:

```
binutils-2.17.50.0.6-2.el5  
compat-libstdc++-33-3.2.3-61  
elfutils-libelf-0.125-3.el5  
elfutils-libelf-devel-0.125  
gcc-4.1.1-52  
gcc-c++-4.1.1-52  
glibc-2.5-12  
glibc-common-2.5-12  
glibc-devel-2.5-12  
glibc-headers-2.5-12  
libaio-0.3.106  
libaio-devel-0.3.106  
libgcc-4.1.1-52  
libstdc++-4.1.1  
libstdc++-devel-4.1.1-52.el5  
make-3.81-1.1  
sysstat-7.0.0
```

Al igual que pasaba con Clusterware, las versiones de los paquetes las podemos comprobar mediante:

```
rpm -q nombre_paquete
```

En algunos casos ha hecho falta la búsqueda de los rpms y su instalación utilizando en este caso comandos como:

```
rpm -i elfutils-libelf-devel-0.125-3.el5
```

Como ya se ha comentado, al final de todo el apartado se mostrará la instalación de todos los paquetes que han sido necesarios.

Ahora veamos los **parámetros del kernel**:

```
semmsl 250  
semms 32000  
semopm 100  
semmni 128  
shmmax 2147483648
```

```
shmmni 4096
shmall 2097152
file-max 512 * PROCESSES
ip_local_port_range mínimo 1024 - máximo 65000
rmem_default 262144
rmem_max 4194304
wmem_default 262144
wmem_max 262144
aio-max-nr máximo 1048576
```

Como se puede comprobar viendo el mismo apartado en la parte de los requisitos de Clusterware sólo cambian los `file-max` y `wmem_max` que se necesita que sea mayor `file-max` y que sea menor `wmem_max`, lo cual hace que no necesite un cambio ya que su valor actual es superior al mínimo necesitado. Para cambiar `file-max` editamos el archivo `/etc/sysctl.conf` y actualizamos sus valores quedando como sigue:

```
kernel.shmall = 2097152
kernel.shmmax = 2147483648
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 512 * PROCESSES
fs.aio-max-nr=1048576
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 4194304
```

Usuarios y grupos requeridos de 11g

Estas configuraciones deberían ser realizadas y testeadas en el caso de realizar una instalación nueva. En el caso de `nod01`, ya existe la base de datos, con lo cual ya tenemos mucha labor hecha.

Ahora, en cuanto a los grupos, sólo hay que añadir uno nuevo en ambos nodos, `asmadmin` para ASM, ya que también vamos a instalarlo. Cabe recordar que, de inicio, tenemos una base de datos en `single-instance` y con almacenamiento en `file system`, por lo que ASM no está instalado y hay que preparar el sistema para él.

El usuario `oracle` debe estar en este grupo:

```
[oracle@nod01 ~]$ groupadd asmadmin
```

Una vez que ya han sido creados todos los grupos debemos hacer una última configuración para el usuario `oracle`. Debemos asegurarnos de que pertenece a todos los grupos que necesitamos:

```
[oracle@nod01 ~]$ usermod -u 500 -g oinstall -G
dba,oper,asmadmin oracle
```

Configuración de la red para 11g

En este punto no hay nada más que configurar ya que todo lo necesario ha sido hecho en la configuración de los requisitos de Clusterware. El **nodo está preparado para trabajar conectado a la red.**

Configuración del almacenamiento para los archivos de base de datos y los archivos de recovery

Las tareas referentes a este punto son explicadas en el apartado Preparación del almacenamiento, donde se da detalles de todas las tareas realizadas tanto para la

configuración del almacenamiento de la base de datos y ASM como del mismo para Clusterware..

Preparación para RAC

En este apartado trataremos, al igual que en los anteriores, todas las tareas que se deben realizar antes de la transformación de la base de datos single-instance en RAC.

Al llegar a las tareas de preinstalación de RAC con todas las necesarias para Clusterware y la base de datos realizadas sólo queda cerciorarse de que los usuarios y grupos están configurados adecuadamente ya que todos los requisitos han sido cubiertos ya. Todo lo necesario en cuanto a conexiones, espacio de disco, software, hardware, etc ya ha sido cubierto.

En el caso de RAC, los grupos y usuarios son los mismos que hemos ido viendo en los apartados anteriores más el añadido del usuario **nobody**. Este usuario es un usuario sin privilegios que debe ser el propietario de los jobs externos (extjob) ejecutables tras la instalación y no es necesaria su configuración, por lo que no la realizamos.

Instalación de todos los paquetes necesarios

Como ya se ha comentado en apartados anteriores, para que fuera más fácil su comprensión mostramos aquí la instalación de todos los paquetes que ha sido necesario instalar tanto para la instalación de la primera base de datos como para la final, Clusterware, ASM y RAC:

```
[root@nodol mnt]# rpm -Uvh */*/binutils-2.*
warning: cdrom/Server/binutils-2.17.50.0.6-
9.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
ID 1e5e0159
Preparing...
##### [100%]
package binutils-2.17.50.0.6-9.el5.i386 is
already installed

[root@nodol mnt]# rpm -Uvh */*/elfutils-libelf-0.*
warning: cdrom/Server/elfutils-libelf-0.137-
3.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
ID 1e5e0159
Preparing...
##### [100%]
package elfutils-libelf-0.137-3.el5.i386 is
already installed

[root@nodol mnt]# rpm -Uvh */*/glibc-2.*i686*
warning: cdrom/Server/glibc-2.5-34.i686.rpm: Header
V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package glibc-2.5-34.i686 is already
installed

[root@nodol mnt]# rpm -Uvh */*/glibc-common-2.*
warning: cdrom/Server/glibc-common-2.5-34.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package glibc-common-2.5-34.i386 is already
installed
```

```
[root@nodol mnt]# rpm -Uvh */*/libaio-0.*
warning: cdrom/Server/libaio-0.3.106-3.2.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package libaio-0.3.106-3.2.i386 is already
installed

[root@nodol mnt]# rpm -Uvh */*/libgcc-4.*
warning: cdrom/Server/libgcc-4.1.2-44.el5.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package libgcc-4.1.2-44.el5.i386 is already
installed

[root@nodol mnt]# rpm -Uvh */*/libstdc++-4.*
warning: cdrom/Server/libstdc++-4.1.2-
44.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
ID 1e5e0159
Preparing...
##### [100%]
package libstdc++-4.1.2-44.el5.i386 is
already installed

[root@nodol mnt]# rpm -Uvh */*/make-3.*
warning: cdrom/Server/make-3.81-3.el5.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package make-3.81-3.el5.i386 is already
installed

[root@nodol mnt]# rpm -Uvh */*/compat-libstdc++-33*
warning: cdrom3/Server/compat-libstdc++-33-3.2.3-
61.i386.rpm: Header V3 DSA signature: NOKEY, key ID
1e5e0159
Preparing...
##### [100%]
1:compat-libstdc++-33
##### [100%]

[root@nodol mnt]# rpm -Uvh */*/elfutils-libelf-devel-
*
warning: cdrom2/Server/elfutils-libelf-devel-0.137-
3.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
ID 1e5e0159
Preparing...
##### [100%]
package elfutils-libelf-devel-static-0.137-
3.el5.i386 is already installed
package elfutils-libelf-devel-0.137-
3.el5.i386 is already installed

[root@nodol mnt]# rpm -Uvh */*/glibc-devel-2.*
warning: cdrom2/Server/glibc-devel-2.5-34.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package glibc-devel-2.5-34.i386 is already
installed

[root@nodol mnt]# rpm -Uvh */*/glibc-headers*
warning: cdrom2/Server/glibc-headers-2.5-34.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package glibc-headers-2.5-34.i386 is already
installed

[root@nodol mnt]# rpm -Uvh */*/libgomp*
```

```
warning: cdrom2/Server/libgomp-4.3.2-7.el5.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package libgomp-4.3.2-7.el5.i386 is already
installed

[root@nodol mnt]# rpm -Uvh */*/gcc-4.*
warning: cdrom2/Server/gcc-4.1.2-44.el5.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package gcc-4.1.2-44.el5.i386 is already
installed

[root@nodol mnt]# rpm -Uvh */*/gcc-c++-4.*
warning: cdrom2/Server/gcc-c++-4.1.2-44.el5.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
package gcc-c++-4.1.2-44.el5.i386 is already
installed

[root@nodol mnt]# rpm -Uvh */*/libaio-devel-0.*
warning: cdrom3/Server/libaio-devel-0.3.106-
3.2.i386.rpm: Header V3 DSA signature: NOKEY, key ID
1e5e0159
Preparing...
##### [100%]
1:libaio-devel
##### [100%]

[root@nodol mnt]# rpm -Uvh */*/libstdc++-devel-4.*
warning: cdrom2/Server/libstdc++-devel-4.1.2-
44.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
ID 1e5e0159
Preparing...
##### [100%]
package libstdc++-devel-4.1.2-44.el5.i386 is
already installed

[root@nodol mnt]# rpm -Uvh */*/unixODBC-2.*
warning: cdrom2/Server/unixODBC-2.2.11-7.1.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
1:unixODBC
##### [100%]

[root@nodol mnt]# rpm -Uvh */*/unixODBC-devel-2.*
warning: cdrom3/Server/unixODBC-devel-2.2.11-
7.1.i386.rpm: Header V3 DSA signature: NOKEY, key ID
1e5e0159
Preparing...
##### [100%]
1:unixODBC-devel
##### [100%]

[root@nodol mnt]# rpm -Uvh */*/sysstat-7.*
warning: cdrom3/Server/sysstat-7.0.2-3.el5.i386.rpm:
Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
##### [100%]
1:sysstat
##### [100%]
```

Preparación del almacenamiento

En este apartado vamos a encargarnos de configurar el almacenamiento para que trabaje de forma perfecta tanto para la base de datos como para Clusterware y, finalmente con ASM.

Hay que tener en cuenta que vamos a realizar una migración del sistema actual a otro y que el actual tiene unas características concretas en cuanto a almacenamiento. La base de datos actual trabaja con **file system storage**, con lo cual vamos a trabajar de esa forma hasta que sea indispensable cambiarlo (con la instalación ASM).

Configuración de discos compartidos

Como se ha comentado en la parte de Conceptos, ASM y Clusterware, tienen un alto grado de complejidad y necesitan de configuraciones específicas.

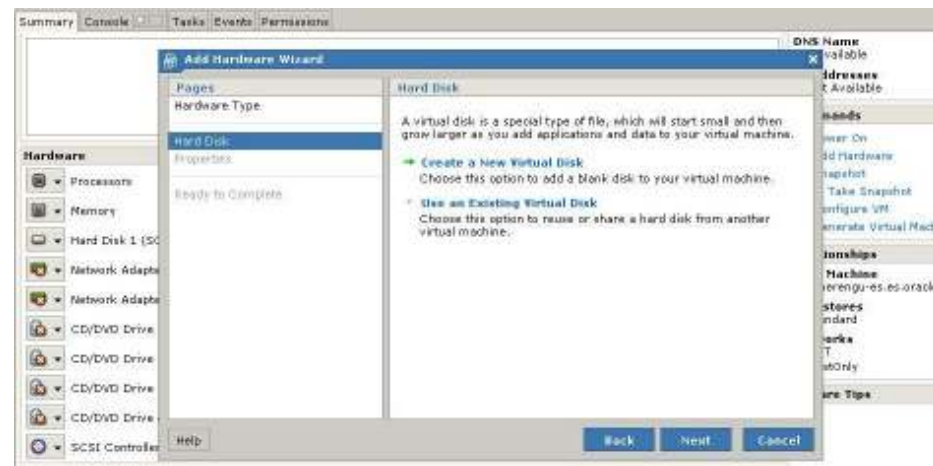
En el caso de Clusterware vamos a configurar dos discos compartidos, uno para su voting disk y otro para su ocr, mientras que para ASM configuraremos dos discos más grandes llamados FRA y DATA, cuya utilización explicaremos más adelante.

Siendo más específicos, los datos de los discos son los siguientes:

SCSI 1:0	0.5 GB	[standard] shared/vote.vmdk
SCSI 1:1	0.5 GB	[standard] shared/ocr.vmdk
SCSI 1:2	4 GB	[standard] shared/data.vmdk
SCSI 1:3	4 GB	[standard] shared/fra.vmdk

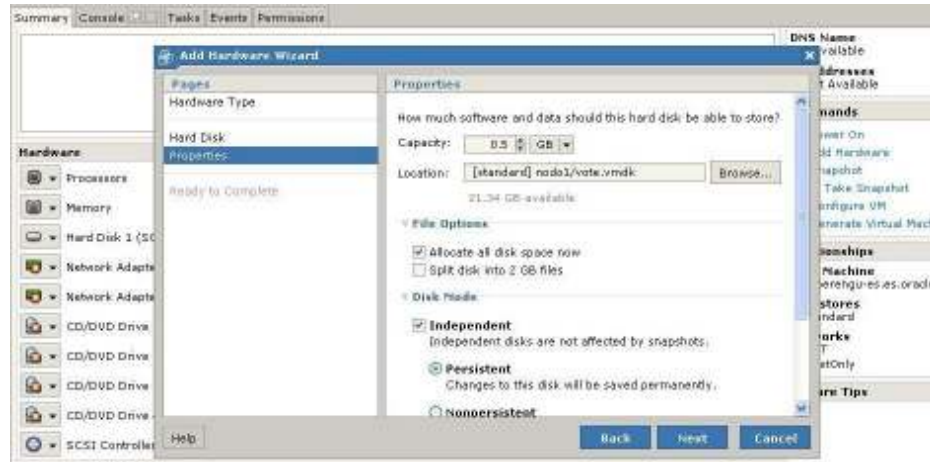
El tamaño de los discos es así ya que OCR y voting disk tienen suficiente con el asignado al no necesitar de muchas respectivas labores, y data y fra serán los discos donde se guardarán los archivos de la base de datos y el Flash Recovery Area respectivamente, con lo que se considera que con 4 GB será suficiente ya que, al fin y al cabo, nuestra base de datos es de test.

Aprovechando que estamos trabajando con vmware vamos a poder configurar estos discos desde la consola de administración de nuestra máquina virtual. Así pues, en el apartado *Commands* clicamos sobre *Add Hardware* y en la pantalla que mostramos a continuación lo hacemos sobre *Create a New Virtual Disk*:

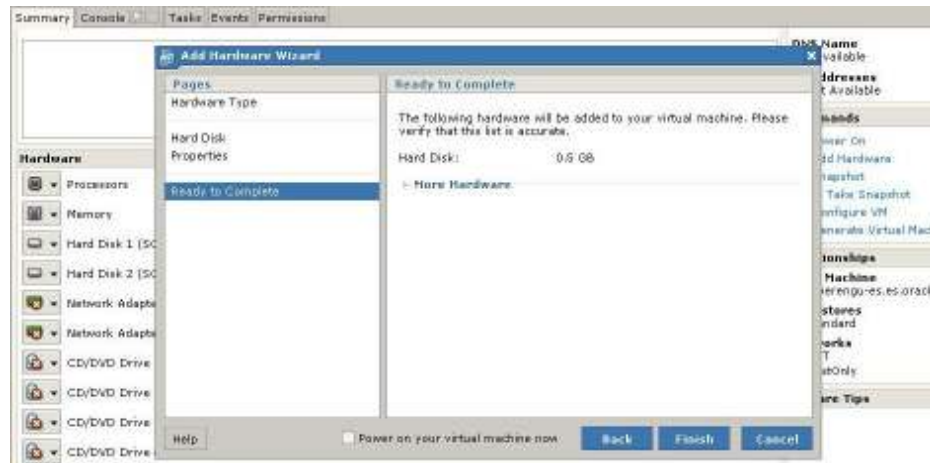


En la siguiente pantalla configuramos las características del disco. El que se muestra ahora es el destinado al voting disk de Clusterware por lo tanto su capacidad es de 0.5 GB, lo localizamos en [standard] shared/vote.vmdk, seleccionamos *Allocate*

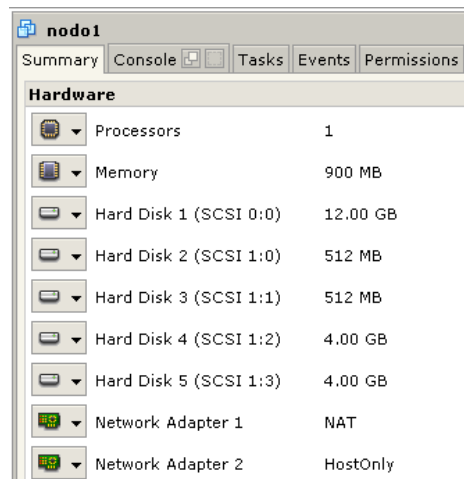
all disk space now, el modo independiente y persistente y le indicamos (aunque no se ve en la captura) el *Virtual Device Node SCSI 1:0*:



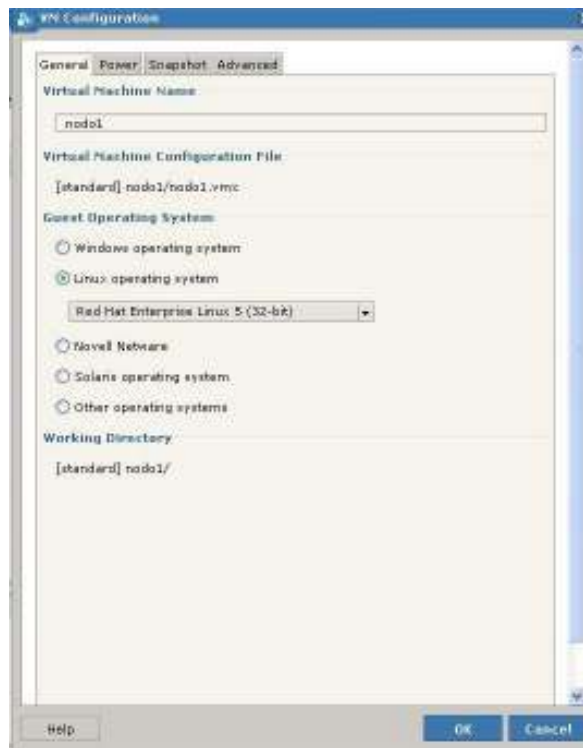
Clicamos en *Finish* para crearlo:



Como hemos indicado antes, debemos configurar cuatro discos. Así pues repetimos esta operación tres veces más y comprobamos en *Summary* que se ha configurado todo correctamente:



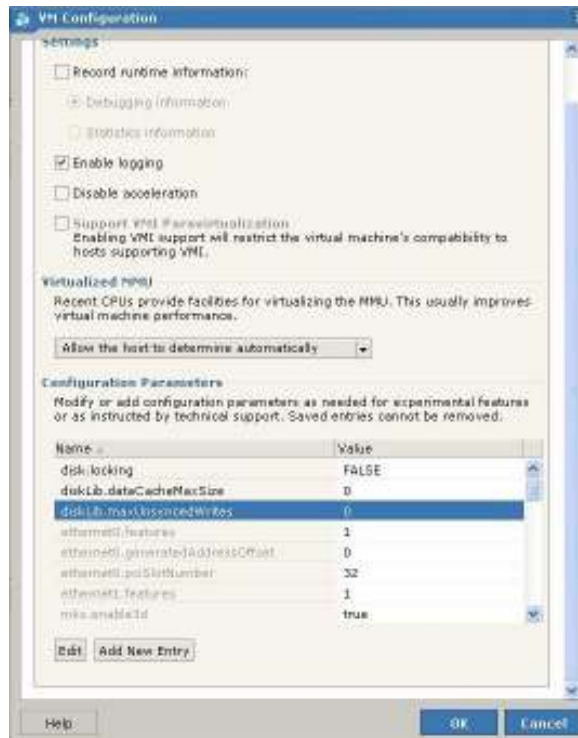
Tras esto continuamos con la configuración de los discos clicando en el apartado *Commands, Configure VM*. Nos aparece la siguiente pantalla, en la cual, pinchamos sobre *Advanced*:



Ahora configuramos los *Configuration Parameters*. A estos vamos a añadir los siguientes con sus respectivos valores:

disk.locking	FALSE
diskLib.dataCacheMaxSize	0
diskLib.maxUnsyncedWrites	0

Utilizamos el botón *Add New Entry* para añadir los parámetros, que quedan como se muestra a continuación, y clicamos en *OK* al terminar:



Una vez creados los discos compartidos reinstanciamos la máquina y comprobamos que podemos acceder a ellos mediante:

```
[oracle@nodol ~]$ cat /proc/partitions
major minor #blocks name

      8     0 12582912 sda
      8     1   104391 sda1
      8     2 12474472 sda2
      8    16   524288 sdb
      8    32   524288 sdc
      8    48  4194304 sdd
      8    64  4194304 sde
    253     0 10616832 dm-0
    253     1  1835008 dm-1
```

Ahora creamos particiones en todos los discos que hemos creado (sdb, sdc, sdd, sdd) mediante **fdisk**. La ejecución para sdb se muestra a continuación. Crearemos una partición nueva, que ocupe todo el espacio disponible y con un formato Non-FS data:

```
[root@nodol ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table,
nor Sun, SGI or OSF disklabel
Building a new DOS disklabel. Changes will remain in
memory only,
until you decide to write them. After that, of
course, the previous
content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4
will be corrected by w(rite)

Command (m for help): n
Command action
   e   extended
```

```
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-512, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-512,
default 512):
Using default value 512

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): da
Changed system type of partition 1 to da (Non-FS
data)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Estas operaciones, como ya hemos comentado, las repetimos para las particiones restantes.

Una vez hechas las particiones en todos los discos debemos dar permisos al usuario oracle sobre el voting disk y el OCR (sdb1 y sdc1). Esto lo hacemos editando el archivo rc.local, situado en /etc/rc.d:

```
[root@nod01 ~]# gedit /etc/rc.d/rc.local
chown oracle:oinstall /dev/sdb1
chown oracle:oinstall /dev/sdc1
chmod 644 /dev/sdb1
chmod 644 /dev/sdc1
```

Tras editarlo lo reejecutamos:

```
[root@nod01 ~]# /etc/rc.d/rc.local
```

Y por último comprobamos los discos, sus particiones y los permisos:

```
[root@nod01 ~]# ls -l /dev/sd*
brw-r----- 1 root disk 8, 0 Dec 16 12:28 /dev/sda
brw-r----- 1 root disk 8, 1 Dec 16 12:29 /dev/sda1
brw-r----- 1 root disk 8, 2 Dec 16 12:28 /dev/sda2
brw-r----- 1 root disk 8, 16 Dec 16 12:54 /dev/sdb
brw-r--r-- 1 oracle oinstall 8, 17 Dec 16 12:54
/dev/sdb1
brw-r----- 1 root disk 8, 32 Dec 16 12:55 /dev/sdc
brw-r--r-- 1 oracle oinstall 8, 33 Dec 16 12:55
/dev/sdc1
brw-r----- 1 root disk 8, 48 Dec 16 13:00 /dev/sdd
brw-r----- 1 root disk 8, 49 Dec 16 13:00 /dev/sdd1
brw-r----- 1 root disk 8, 64 Dec 16 13:01 /dev/sde
brw-r----- 1 root disk 8, 65 Dec 16 13:01 /dev/sde1
```

Configuración del almacenamiento para la base de datos y ASM

Como sabemos, ASM va a ser el sistema de almacenamiento de la base de datos tras la migración, es decir, cuando hablemos de la configuración del almacenamiento de la base de datos estaremos hablando de ASM.

En esta sección vamos a configurar los discos para utilizarlos con Automatic Storage Management (ASM).

En el nuevo sistema que estamos configurando vamos a utilizar ASM para los datafiles y los recovery files. Además queremos utilizar una **redundancia externa** ya que es lo más recomendado para el sistema que tenemos configurado (máquina virtual). En un grupo de discos con redundancia normal, para incrementar el rendimiento y la fiabilidad, ASM utiliza por defecto two-way mirroring. Un grupo de discos con redundancia normal requiere de **un mínimo de dos discos** (o dos grupos de fallo). El espacio efectivo de disco en un grupo de discos con redundancia normal es la mitad de todo el espacio de disco de los dispositivos.

Hay que tener en cuenta varios puntos a la hora de elegir y configurar los dispositivos de disco adecuados:

- Es obligatorio que todos los dispositivos en el grupo de discos de ASM sean del mismo tamaño y tengan las mismas características de rendimiento. Los discos que hemos creado son exactamente iguales teniendo en cuenta esto.
- No se debe especificar más de una partición en un disco como dispositivo de grupo de discos. ASM espera que cada dispositivo de grupo de discos esté en discos físicos diferentes.
- No es recomendable utilizar volúmenes lógicos ya que puede provocar problemas con la arquitectura física de los discos y reducir la optimización que hace ASM del I/O en los dispositivos. Sin embargo, **vamos a utilizar discos lógicos** por la misma razón que hemos comentado ya otras veces, nuestro sistema está configurado en máquinas virtuales.

Oracle proporciona un driver de librería de ASM, **oracleasmlib**, que simplifica la configuración y administración de los dispositivos de disco que se quiere utilizar con ASM. Un disco configurado para ASM se le conoce como disco *candidato*. Esto lo podremos comprobar en la instalación.

Configuración del almacenamiento para los archivos de base de datos

Esta tarea debe ser realizada después de la instalación y configuración del almacenamiento de Clusterware y después de haber revisado las opciones de almacenamiento de disco para los archivos de base de datos. Aunque la realizaremos después de instalar Clusterware y el software de la base de datos 11g lo comentamos aquí.

Esta configuración se debe hacer en dos pasos:

1. Chequeo del almacenamiento compartido disponible con CVU.

Para realizar una comprobación de todos los file systems compartidos disponibles por todos los nodos del cluster es recomendable ejecutar la herramienta runcluvfy (CVU). Sin embargo, como nosotros hemos configurado a propósito los discos y sabemos de su estado este chequeo lo podemos obviar.

2. Configurar el almacenamiento para los archivos de base de datos y los archivos de recovery.

Esta tarea consiste en configurar los discos que vayan a ser utilizados para almacenar los archivos de la base de datos y el recovery para que sean administrados por ASM. Para ello deberemos instalar y configurar **ASM Library Driver**:

Para realizar esta configuración realizaremos los siguientes pasos logueados como root:

1. Instalar y configurar ASMLIB.

Nos situamos en /mnt y ejecutamos los siguientes comandos para la instalación de los rpms necesarios:

```
[root@nod01 ~]# rpm -Uvh */*/oracleasm-support*
warning: cdrom3/Server/oracleasm-support-2.1.2-1.el5.i386.rpm: Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
#####
[100%]
1:oracleasm-support
#####
[100%]

[root@nod01 ~]# rpm -Uvh */*/oracleasm*el5-2*
warning: cdrom3/Server/oracleasm-2.6.18-128.el5-2.0.5-1.el5.i686.rpm: Header V3 DSA signature: NOKEY, key ID 1e5e0159
Preparing...
#####
[100%]
1:oracleasm-2.6.18-128.el5#####
##### [100%]
```

Ahora nos situamos en /home/oracle/setups e instalamos la librería **oracleasmlib**:

```
[root@nod01 setups]# rpm -Uvh oracleasmlib*
Preparing...
#####
[100%]
1:oracleasmlib
#####
[100%]
```

Ejecutamos el siguiente comando para **arrancar el script de inicialización de oracleasm** con la opción de configuración:

```
[root@nod01 ~]# /etc/init.d/oracleasm configure
```

En las siguientes peticiones de información donde se nos solicita el usuario, grupo por defecto y el tipo de arranque introducimos **oracle**, **dba** e **y**:

```
Configuring the Oracle ASM library driver.

This will configure the on-boot properties of
the Oracle ASM library
driver. The following questions will determine
whether the driver is
loaded on boot and what permissions it will
have. The current values
will be shown in brackets ('[]'). Hitting
<ENTER> without typing an
answer will keep that current value. Ctrl-C
will abort.
```

```
Default user to own the driver interface []:  
  oracle  
Default group to own the driver interface []:  
  dba  
Start Oracle ASM library driver on boot (y/n)  
  [n]: y  
Fix permissions of Oracle ASM disks on boot  
  (y/n) [y]: y  
Writing Oracle ASM library driver  
  configuration: done  
Initializing the Oracle ASMLib driver: [ OK ]  
Scanning the system for Oracle ASMLib disks:  
  [ OK ]
```

El script que hemos ejecutado realiza las siguientes tareas:

- Crea el archivo de configuración /etc/sysconfig/oracleasm
- Crea el punto de montaje /dev/oracleasm
- Carga el módulo de kernel oracleasm
- Monta el sistema de ficheros de ASMLIB driver. Este sistema de ficheros no es un file system regular. Sólo es utilizado por la librería de ASM para comunicarse con el driver de ASM.

Este proceso debemos **repetirlo también en el segundo nodo**.

2. Configurar los dispositivos de disco para utilizar ASMLIB.

Hasta ahora la base de datos ha trabajado con SCSI así que continuaremos con ello ya que el cambio de almacenamiento se hará después de la migración.

Para realizar la configuración de los discos debemos tener en cuenta que, según nuestra configuración actual, los discos ya están compartidos y que, por lo tanto, sólo necesitamos inicializarlos.

Ejecutamos:

```
[root@nod01 ~]# /etc/init.d/oracleasm  
  createdisk fra /dev/sdd1  
Marking disk "fra" as an ASM disk: [ OK ]  
[root@nod01 ~]# /etc/init.d/oracleasm  
  createdisk data /dev/sde1  
Marking disk "data" as an ASM disk: [ OK ]  
[root@nod01 ~]# /etc/init.d/oracleasm listdisks  
FRA  
DATA
```

3. Administrar ASMLIB y los discos.

Una vez hecho todo el proceso sólo nos quedaría, en el caso de que fuera necesario, la administración de los discos y de la librería. Esto se puede hacer mediante el comando que hemos ejecutado al principio para arrancar el script de inicialización de oracleasm con diferentes opciones:

- Createdisk para crear y añadir un disco nuevo
- Configure para configurar los discos y ASMLIB
- Deletedisk para eliminar un disco del grupo
- Scandisks para que los nodos identifiquen qué discos compartidos han sido marcados como discos de ASM en otros nodos.

Configuración del almacenamiento para RAC

En RAC hay tres opciones de almacenamiento:

- Automatic Storage Management (**ASM**) que es la que vamos a utilizar.
- File System compartido y soportado
- Block o Raw devices

Tal como se indica en el objetivo de este proyecto, se quiere migrar una base de datos de Single Instance a RAC con **ASM**. Por lo tanto, la decisión de elegir ASM es obvia. Además, a día de hoy es la opción recomendada ya que ASM es un sistema de archivos y administrador de discos que permite un alto rendimiento además de striping y mirroring de los archivos de la base de datos automáticamente.

En el caso de Clusterware debíamos buscar un sistema de almacenamiento adecuado para sus archivos, archivos de OCR y voting disk; en el caso de RAC los archivos de RAC son los propios de la base de datos y los de recovery, a los cuales ASM da soporte totalmente.

Una vez hecha la elección de almacenamiento de discos y después de haber instalado y configurado el almacenamiento de Clusterware y, también después de haber revisado las opciones de almacenamiento de disco para la base de datos, se debería realizar las siguientes tareas:

1. Hacer un chequeo del almacenamiento compartido disponible con CVU.
2. Configurar el almacenamiento para los archivos de la base de datos y los archivos de recovery.

Estas tareas, como se puede comprobar, son en resumidas cuentas las tareas de configuración de ASM y **ya han sido explicadas en el punto anterior**.

Ejecución de la migración

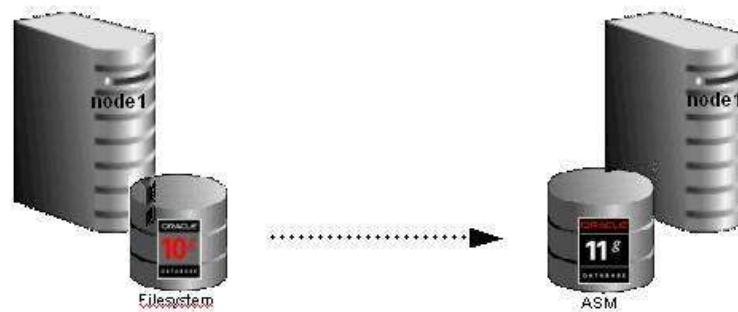
Antes de iniciar el proceso de migración en sí del sistema se debe haber terminado con éxito todas las tareas anteriores y se debe tener constancia de que todo está en las condiciones que se ha especificado.

Como se ha explicado en el primer apartado de este documento, el objetivo es realizar la migración de una base de datos en single-instance (una sola instancia) 10.2.0.4 a una base de datos en RAC de un nodo 11.1.0.6 realizando también el paso de file system a ASM. Para conseguir este objetivo se va a dividir todo el trabajo en varios pasos para una mejor comprensión.

Los pasos de que consta el proceso son:

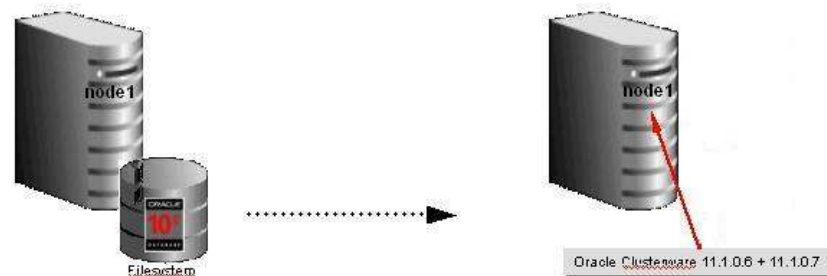
1. Instalar Clusterware 11.1.0.6
2. Instalar los clustered homes 11.1.0.6 para la base de datos y ASM y crear la instancia ASM con cluster 11.1.0.6
3. Upgrade de single instance 10.2.0.4 a single instance 11.1.0.6
4. Cambiar la single instance 11.1.0.6 a ASM con RMAN
5. Convertir la single instance 11.1.0.6 en una base de datos RAC 11.1.0.6

Para poder observar el **objetivo final** de una forma más clara se muestra el siguiente gráfico:



Instalar Clusterware 11.1.0.6

Como ya tenemos todo preparado para empezar nuestro primer paso de la migración es la instalación de Clusterware 11.1.0.6.



Antes de la instalación de Clusterware es un paso obligatorio ejecutar la herramienta **runcluvfy**. Esta herramienta nos permite hacer un chequeo del sistema operativo y del hardware antes de realizar la instalación para asegurarnos de cumplir todas las

exigencias. Evidentemente, todo lo necesario lo hemos realizado ya en el apartado **Premigración**, pero es obligatorio la ejecución en el momento antes de iniciar la instalación.

Nos situamos en el directorio donde se encuentra el software que vamos a instalar, /home/oracle/setups/clusterware y ejecutamos:

```
[oracle@nod01 clusterware]$ ./runcluvfy.sh stage -
post hwos -n nod01
Performing post-checks for hardware and operating
system setup

Checking node reachability...
Node reachability check passed from node "nod01".

Checking user equivalence...
User equivalence check passed for user "oracle".

Checking node connectivity...

Node connectivity check passed for subnet
"192.168.201.0" with node(s) nod01.
Node connectivity check passed for subnet
"192.168.160.0" with node(s) nod01.

Interfaces found on subnet "192.168.201.0" that are
likely candidates for a private interconnect:
nod01 eth0:192.168.201.128

Interfaces found on subnet "192.168.160.0" that are
likely candidates for a private interconnect:
nod01 eth1:192.168.160.128

WARNING:
Could not find a suitable set of interfaces for VIPs.

Node connectivity check passed.

Checking shared storage accessibility...

WARNING:
Package cvuqdisk not installed.
nod01

Shared storage check failed on nodes "nod01".

Post-check for hardware and operating system setup
was unsuccessful on all the nodes.
```

Debido a nuestra configuración mediante Vmware el **test de memoria compartida** ha dado error. Esto no es un problema, podemos continuar con la instalación.

A continuación realizamos el último chequeo sobre los servicios del cluster para la instalación de Clusterware utilizando también **runcluvfy**. Ejecutamos:

```
[oracle@nod01 clusterware]$ ./runcluvfy.sh stage -pre
crsinst -n nod01 -r 11gR1

Performing pre-checks for cluster services setup

Checking node reachability...
Node reachability check passed from node "nod01".
```



```
Checking user equivalence...
User equivalence check passed for user "oracle".

Checking administrative privileges...
User existence check passed for "oracle".
Group existence check passed for "oinstall".
Membership check for user "oracle" in group
  "oinstall" [as Primary] passed.

Administrative privileges check passed.

Checking node connectivity...

Node connectivity check passed for subnet
  "192.168.201.0" with node(s) nodol.
Node connectivity check passed for subnet
  "192.168.160.0" with node(s) nodol.

Interfaces found on subnet "192.168.201.0" that are
  likely candidates for a private interconnect:
nodol eth0:192.168.201.128

Interfaces found on subnet "192.168.160.0" that are
  likely candidates for a private interconnect:
nodol eth1:192.168.160.128

WARNING:
Could not find a suitable set of interfaces for VIPs.

Node connectivity check passed.

Checking system requirements for 'crs'...
Total memory check failed.
Check failed on nodes:
  nodol
Free disk space check passed.
Swap space check passed.
System architecture check passed.
Kernel version check passed.
Package existence check passed for "make-3.81".
Package existence check passed for "binutils-
  2.17.50.0.6".
Package existence check passed for "gcc-4.1.1".
Package existence check passed for "libaio-0.3.106".
Package existence check passed for "libaio-devel-
  0.3.106".
Package existence check passed for "libstdc++-4.1.1".
Package existence check passed for "elfutils-libelf-
  devel-0.125".
Package existence check passed for "sysstat-7.0.0".
Package existence check passed for "compat-libstdc++-
  33-3.2.3".
Package existence check passed for "libgcc-4.1.1".
Package existence check passed for "libstdc++-devel-
  4.1.1".
Package existence check passed for "unixODBC-2.2.11".
Package existence check passed for "unixODBC-devel-
  2.2.11".
Package existence check passed for "glibc-2.5-12".
Group existence check passed for "dba".
Group existence check passed for "oinstall".
User existence check passed for "nobody".

System requirement failed for 'crs'
```

```
Pre-check for cluster services setup was unsuccessful  
on all the nodes.
```

Este último chequeo muestra errores en el total de la memoria y el espacio de swap. Al igual que pasaba antes, al estar realizando la instalación en un entorno Vmware que, aunque está configurado prácticamente como un entorno de producción, no lo es al 100%, estos fallos de memoria y swap pueden ser ignorados.

Una vez hechos ya los últimos chequeos pasamos ya a la instalación de Clusterware mediante el OUI (Oracle Universal Installer).

En el directorio donde nos encontramos arrancamos el OUI, que está situado en este mismo directorio:

```
[oracle@nod01 clusterware]$ ./runInstaller
```

A continuación se muestra el interfaz gráfico de instalación.



En esta primera pantalla simplemente clicaremos sobre *Next* y nos aparecerá una nueva en la que se nos pide que indiquemos el **path de oraInventory**, directorio que hemos creado anteriormente, y el nombre del **grupo del sistema operativo** que va a realizar la ejecución, **oinstall**, que también hemos configurado anteriormente. Lo indicamos de la siguiente manera:



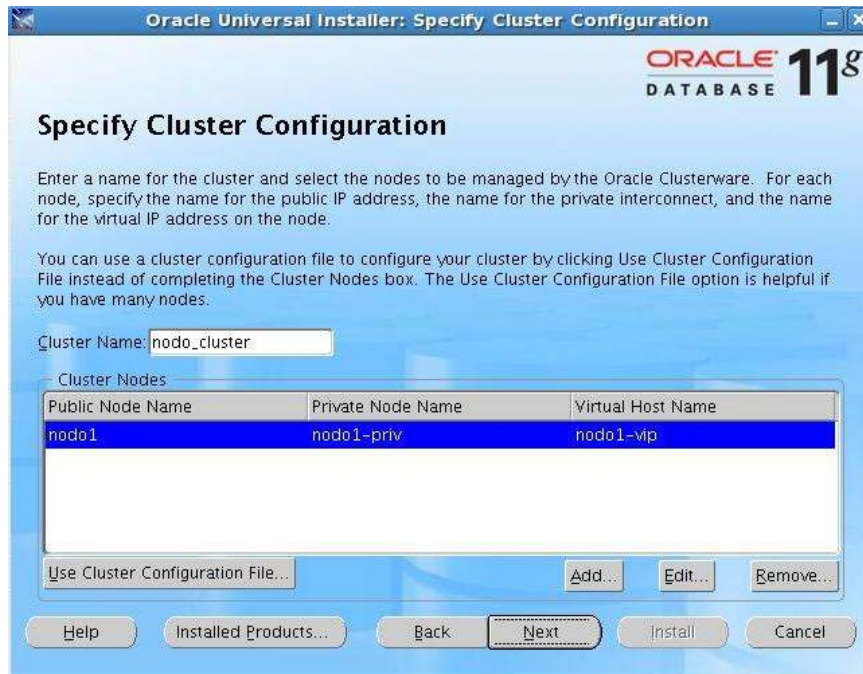
En la siguiente pantalla se nos pide que indiquemos un nombre para la instalación, `OraCrs11g_home`, y el path donde queremos que se realice, `/u01/crs/oracle/product/11.1.0/crs`:



En la siguiente pantalla el instalador realiza un chequeo de los requisitos específicos del producto. Como se ve en la captura abajo **se cumple con todo menos un warning para el test de memoria física**. Lo marcamos como *User verified* (ya sabíamos que la memoria no cumplía todos los requisitos) y continuamos.



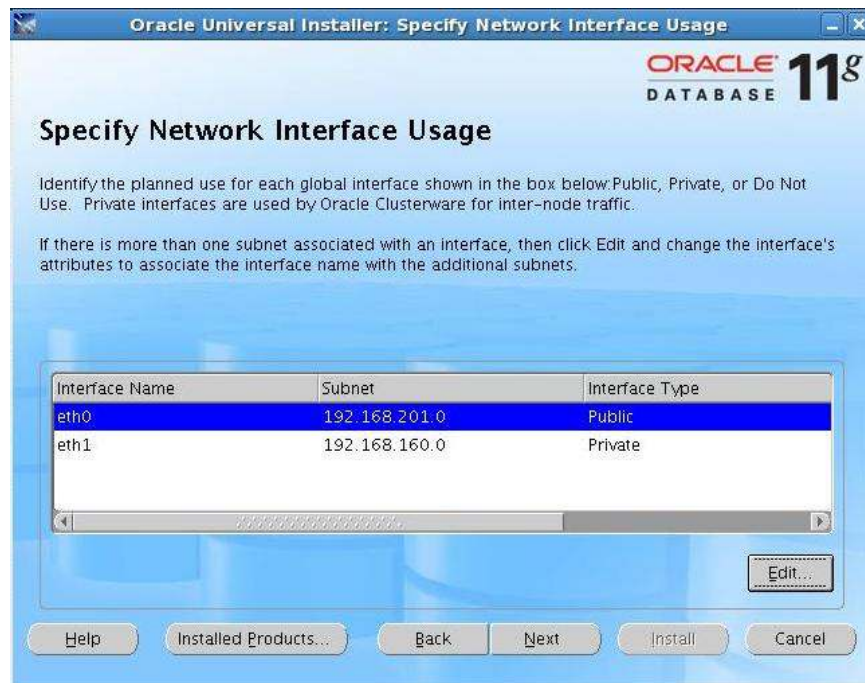
La siguiente pantalla es para la configuración del cluster. Aquí debemos indicar el nombre del cluster, `nodo_cluster`, y los nodos que queremos que formen parte del cluster, en nuestro caso únicamente el local, `nodo1`. Comprobamos que los datos del **nombre público**, **nombre privado** y **nombre del host virtual** son correctos y clicamos sobre *Next*.



Ahora debemos introducir en la nueva pantalla los datos referentes a las interfaces de red que queremos utilizar en el cluster. Aparecen ya las que tenemos configuradas en el `nodo1`, `eth0` y `eth1`. Sólo debemos indicar cual queremos que sea la pública y cual la privada. Elegimos **eth0** como **pública** y **eth1** como **privada**:



La configuración final de las interfaces se muestra a continuación:



En la nueva pantalla de **configuración del Oracle Cluster Registry (OCR)**, al realizar todo el trabajo mediante Vmware, elegimos **redundancia externa** ya que no vamos a utilizar mirroring de OCR.

Localizamos el OCR en /dev/sdc1:



En la siguiente pantalla se acomete la configuración del Voting Disk de Clusterware. Al igual que con OCR, elegimos redundancia externa ya que no vamos a utilizar varios discos para almacenar todas las copias del voting disk.

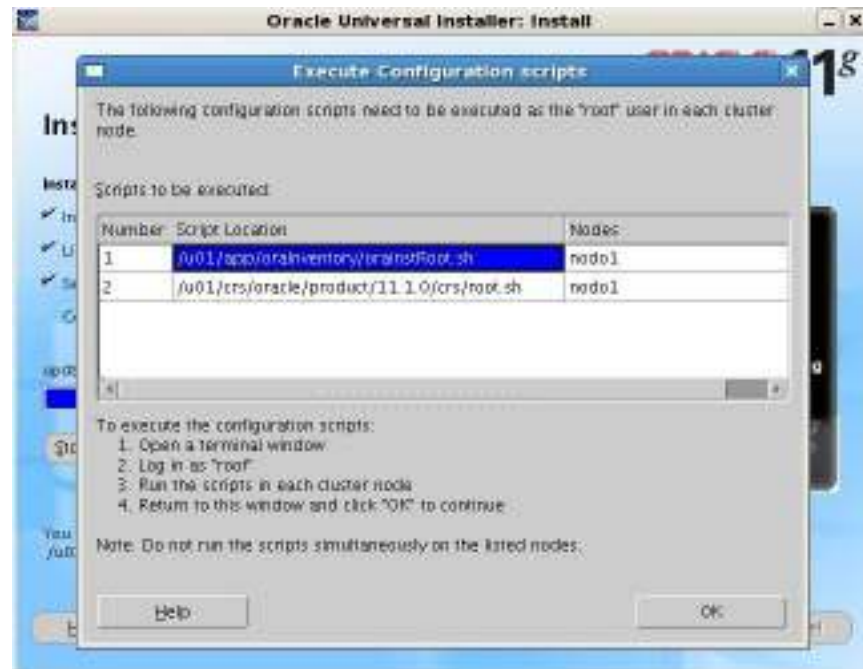
Almacenamos el **Voting Disk** en /dev/sdb1:



En la siguiente pantalla se nos muestra todas las configuraciones y productos que se van a instalar finalmente. Clicando en *Install* empieza definitivamente la instalación, la cual hasta el final ya no requerirá de nuestra participación:



Una vez terminada la instalación se nos pide que ejecutemos como `root` los archivos que se ven en la captura en ambos nodos. Estos ficheros terminan de preparar la configuración del cluster y automáticamente comienza a funcionar.



A continuación mostramos el resultado de la ejecución de ambos scripts. Como se puede observar el primero realiza cambios en los permisos de los directorios y el segundo realiza configuraciones sobre el OCR, CRS stack, añade demonios y crea la VIP entre otras cosas:

```
[root@nod01 ~]# /u01/app/orainventory/orainstRoot.sh
Changing permissions of /u01/app/orainventory to 770.
Changing groupname of /u01/app/orainventory to
oinstall.
```

The execution of the script is complete

```
[root@nod01 ~]#
/u01/crs/oracle/product/11.1.0/crs/root.sh
WARNING: directory '/u01/crs/oracle/product/11.1.0'
is not owned by root
WARNING: directory '/u01/crs/oracle/product' is not
owned by root
WARNING: directory '/u01/crs/oracle' is not owned by
root
WARNING: directory '/u01/crs' is not owned by root
WARNING: directory '/u01' is not owned by root
Checking to see if Oracle CRS stack is already
configured
/etc/oracle does not exist. Creating it now.

Setting the permissions on OCR backup directory
Setting up Network socket directories
Oracle Cluster Registry configuration upgraded
successfully
The directory '/u01/crs/oracle/product/11.1.0' is not
owned by root. Changing owner to root
The directory '/u01/crs/oracle/product' is not owned
by root. Changing owner to root
The directory '/u01/crs/oracle' is not owned by root.
Changing owner to root
The directory '/u01/crs' is not owned by root.
Changing owner to root
The directory '/u01' is not owned by root. Changing
owner to root
Successfully accumulated necessary OCR keys.
Using ports: CSS=49895 CRS=49896 EVMC=49898 and
EVMR=49897.
node <nodenumber>: <nodename> <private interconnect
name> <hostname>
node 1: nod01 nod01-priv nod01
Creating OCR keys for user 'root', privgrp 'root'..
Operation successful.
Now formatting voting device: /dev/sdb1
Format of 1 voting devices complete.
Startup will be queued to init within 30 seconds.
Adding daemons to inittab
Expecting the CRS daemons to be up within 600
seconds.
Cluster Synchronization Services is active on these
nodes.
    nod01
Cluster Synchronization Services is active on all the
nodes.
Waiting for the Oracle CRSD and EVMD to start
Oracle CRS stack installed and running under init(1M)
Running vipca(silent) for configuring nodeapps

Creating VIP application resource on (1) nodes..
Creating GSD application resource on (1) nodes..
Creating ONS application resource on (1) nodes..
Starting VIP application resource on (1) nodes1:CRS-
1006: No more members to consider
CRS-0215: Could not start resource 'ora.nod01.vip'.
Check the log file
"/u01/crs/oracle/product/11.1.0/crs/log/nod01/racg/o
ra.nod01.vip.log" for more details
..
Starting GSD application resource on (1) nodes..
Starting ONS application resource on (1) nodes..
```


Done.

Se puede comprobar que cuando se intenta arrancar la VIP genera un **error CRS-0215**. Cómo hemos solucionado este error es mostrado en el apartado **Problemas encontrados** en el apartado **Fallo en la configuración del virtual host**.

A continuación vamos a realizar unos pasos que no son necesarios para un entorno de test como es el nuestro, pero son interesantes de cara a posibles futuras configuraciones en entornos de producción.

Logueados como `root` **configuramos el módulo de kernel `hangcheck-timer`**. Oracle recomienda settings mucho más agresivos como se indica en el Installation Guide de CRS y la nota 726833.1, pero nosotros utilizaremos valores conservadores para nuestro entorno de test en VMware. Ejecutamos:

```
[root@nodol ~]# echo modprobe hangcheck-timer
hangcheck_tick=60 hangcheck_margin=600
hangcheck_reboot=1 >>/etc/rc.d/rc.local

[root@nodol ~]# modprobe hangcheck-timer
hangcheck_tick=60 hangcheck_margin=600
hangcheck_reboot=1
```

Bajamos el CRS:

```
[root@nodol ~]# cd
/u01/crs/oracle/product/11.1.0/crs/bin/

[root@nodol bin]# ./crsctl stop crs

Stopping resources.
This could take several minutes.
Successfully stopped Oracle Clusterware resources
Stopping Cluster Synchronization Services.
Shutting down the Cluster Synchronization Services
daemon.
Shutdown request successfully issued.

[root@nodol bin]# ./oproc stop

Dec 17 12:36:23.676 | ERR | failed to connect to
daemon, errno(111)
```

Configuramos el parámetro `diagwait` para permitir la información de depuración para acceder al disco antes de que `oproc` reinicie un nodo. El valor que vamos a configurar es más alto del recomendado por Oracle en la Installation Guide de CRS y la nota 559365.1).

```
[root@nodol bin]# ./crsctl set css diagwait 33 -force
Configuration parameter diagwait is now set to 33.
```

Reiniciamos el crs:

```
[root@nodol bin]# ./crsctl start crs
Attempting to start Oracle Clusterware stack
The CRS stack will be started shortly
```

Esperamos un minute (el crs tarda en reiniciarse) y realizamos un test de su estado:

```
[root@nodol bin]# ./crsctl check crs
Cluster Synchronization Services appears healthy
Cluster Ready Services appears healthy
Event Manager appears healthy
```

Ahora **incrementamos el `disktimeout`** ya que, según indica la nota 284752.1, VMware puede tardar en arrancar en algunos portátiles (como es el caso).

```
[root@nodol bin]# ./crsctl set css disktimeout 600
Configuration parameter disktimeout is now set to
600.
```

Con esto damos por concluida la instalación y configuración de Clusterware.

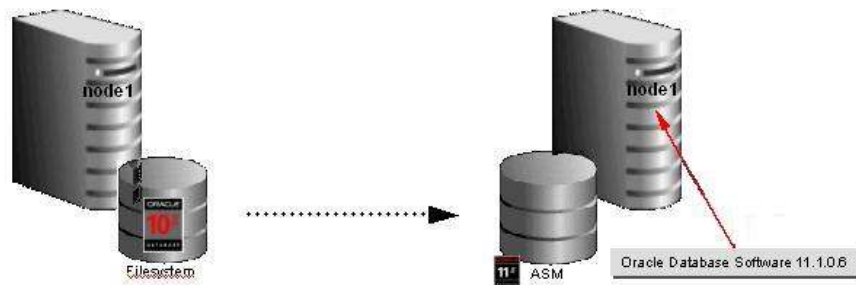
Instalar los clustered homes 11.1.0.6 para la base de datos y ASM y crear la instancia ASM con cluster 11.1.0.6

En este paso de la ejecución de la migración vamos a realizar las instalaciones del software tanto de la base de datos 11.1.0.6 como de ASM para luego crear la instancia de ASM para el cluster que ya hemos creado.

El software de la base de datos debe ser instalado primero ya que sobre éste se asienta ASM. Además necesitamos que el entorno esté con el software disponible y configurado para, en un paso siguiente, realizar el cambio de versión de la base de datos.

Así pues, los pasos que vamos a seguir en este punto son:

1. Instalación del software de la base de datos 11.1.0.6
2. Instalación de ASM y creación de la instancia de ASM



Instalación del software de la base de datos

Al igual que hicimos antes de la instalación de Clusterware, ejecutamos **runcluvfy** para comprobar que nuestro **sistema cumple con todos los requerimientos** necesarios para la instalación del software de la base de datos:

```
[oracle@nod01 clusterware]$ ./runcluvfy.sh stage -pre  
dbinst -n nod01 -r 11gR1
```

```
Performing pre-checks for database installation
```

```
Checking node reachability...
```

```
Node reachability check passed from node "nod01".
```

```
Checking user equivalence...
```

```
User equivalence check passed for user "oracle".
```

```
Checking administrative privileges...
```

```
User existence check passed for "oracle".
```

```
Group existence check passed for "oinstall".
```

```
Membership check for user "oracle" in group
```

```
"oinstall" [as Primary] passed.
```

```
Group existence check passed for "dba".
```

```
Membership check for user "oracle" in group "dba"  
passed.
```

```
Administrative privileges check passed.
```

```
Checking node connectivity...
```

Node connectivity check passed for subnet
"192.168.201.0" with node(s) nodol.
Node connectivity check passed for subnet
"192.168.160.0" with node(s) nodol.

Interfaces found on subnet "192.168.201.0" that are
likely candidates for a private interconnect:
nodol eth0:192.168.201.128

Interfaces found on subnet "192.168.160.0" that are
likely candidates for a private interconnect:
nodol eth1:192.168.160.128

WARNING:
Could not find a suitable set of interfaces for VIPs.

Node connectivity check passed.

Checking system requirements for 'database'...
Total memory check failed.
Check failed on nodes:
 nodol
Free disk space check passed.
Swap space check passed.
System architecture check passed.
Kernel version check passed.
Package existence check passed for "make-3.81".
Package existence check passed for "binutils-
2.17.50.0.6".
Package existence check passed for "gcc-4.1.1".
Package existence check passed for "libaio-0.3.106".
Package existence check passed for "libaio-devel-
0.3.106".
Package existence check passed for "libstdc++-4.1.1".
Package existence check passed for "elfutils-libelf-
devel-0.125".
Package existence check passed for "sysstat-7.0.0".
Package existence check passed for "compat-libstdc++-
33-3.2.3".
Package existence check passed for "libgcc-4.1.1".
Package existence check passed for "libstdc++-devel-
4.1.1".
Package existence check passed for "unixODBC-2.2.11".
Package existence check passed for "unixODBC-devel-
2.2.11".
Package existence check passed for "glibc-2.5-12".
Kernel parameter check passed for "semmsl".
Kernel parameter check passed for "semmsn".
Kernel parameter check passed for "semopm".
Kernel parameter check passed for "semmsi".
Kernel parameter check passed for "shmmax".
Kernel parameter check passed for "shmall".
Kernel parameter check passed for "shmmni".
Kernel parameter check passed for "file-max".
Kernel parameter check passed for "rmem_default".
Kernel parameter check passed for "rmem_max".
Kernel parameter check passed for "wmem_default".
Kernel parameter check passed for "wmem_max".
Group existence check passed for "dba".
User existence check passed for "nobody".

System requirement failed for 'database'

Checking CRS integrity...

```
Checking daemon liveness...  
Liveness check passed for "CRS daemon".
```

```
Checking daemon liveness...  
Liveness check passed for "CSS daemon".
```

```
Checking daemon liveness...  
Liveness check passed for "EVM daemon".
```

```
Checking CRS health...  
CRS health check passed.
```

```
CRS integrity check passed.
```

```
Pre-check for database installation was unsuccessful  
on all the nodes.
```

Los chequeos de memoria y swap fallan, como se puede observar, pero como ya comentamos también para el caso de Clusterware, podemos obviar estos fallos ya que debido a la configuración de nuestro sistema en Vmware es normal.

El siguiente paso es ya la instalación, que mostramos a continuación:



Después de pasar la pantalla de bienvenida clicando en Next aparece la segunda en la que se nos pide que indiquemos el tipo de instalación que deseamos realizar. En nuestro caso es **Enterprise Edition** ya pretendemos emular un sistema real de producción y con los otros dos tipos no tendríamos todas las funcionalidades disponibles.



En la siguiente pantalla debemos indicar la localización en la que queremos instalar el software. En apartados anteriores ya hemos comentado los directorios creados para las diferentes instalaciones. En este caso indicamos como **ORACLE_BASE** /u01/app/oracle (siempre es esta) y como **home del software** de la base de datos /u01/app/oracle/product/11.1.0/db_1. El **nombre** que le damos es Nod011g_home2.



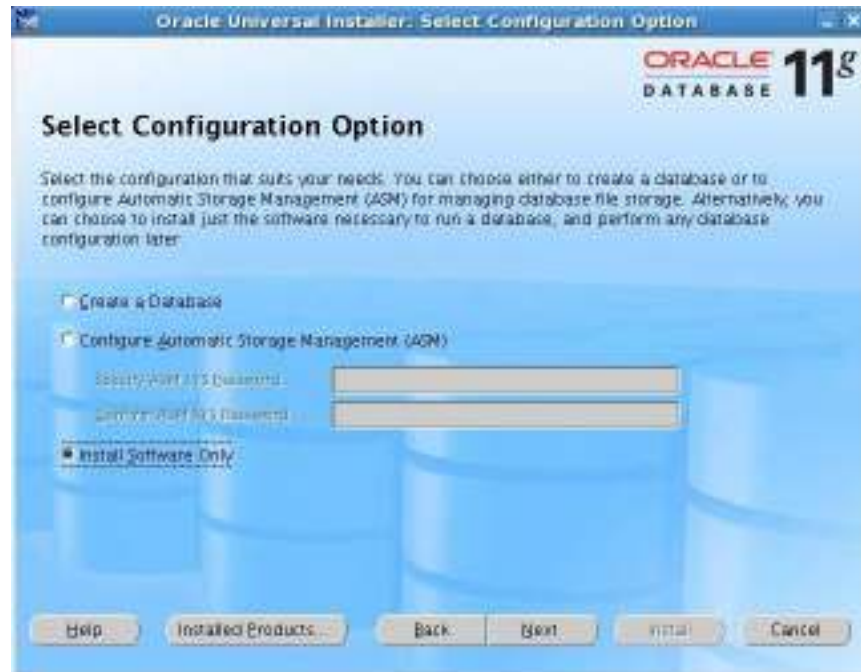
En la siguiente pantalla indicamos la opción de **instalar en Cluster** dejando así la puerta abierta a incrementar posteriormente nuestro cluster con nuevos nodos e instancias. Seleccionamos el **nodo1**, local, que ya aparece seleccionado y es donde estamos realizando la instalación.



La siguiente pantalla muestra el **chequeo de los requisitos específicos del producto**. Como hemos comprobado ejecutando `runcluvfy`, los requisitos de memoria fallan, cosa que obviamos marcándolo como *User Verified*.



En la siguiente pantalla nos aparecen 3 opciones de configuración: crear una base de datos, configurar ASM (cosa que haremos posteriormente) e **instalar únicamente el software de la base de datos**. Seleccionamos esta última opción ya que, como hemos comentado antes, sólo queremos instalar el software para luego realizar el movimiento de la base de datos 10.2.0.4 a esta nueva 11.1.0.6.



En esta última pantalla aparece todo lo que se va a instalar. Clicamos en *Install* y empieza la instalación.



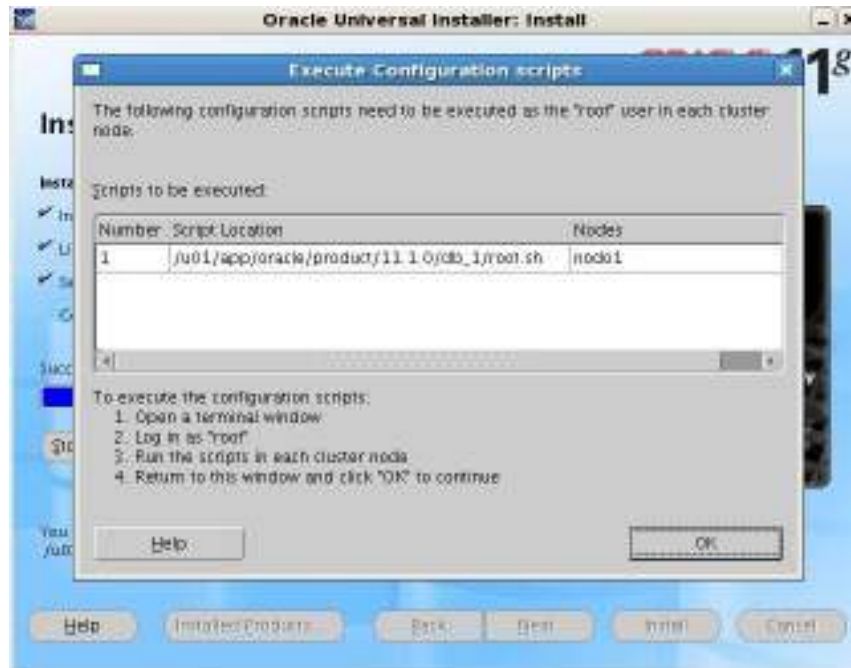
Una vez iniciada la instalación se nos muestra una pantalla de seguimiento de la misma:



Transcurrido un tiempo de la instalación aparece la pantalla del **asistente de configuración** que, mediante el Oracle Net Configuration Assistant y el Oracle Database Configuration Assistant, realiza la configuración del listener de la base de datos y dispone el software de Oracle para poder ser utilizado.



Una vez han terminado con éxito las tareas del asistente de configuración nos aparece la siguiente pantalla en la que se nos indica que debemos ejecutar el script `root.sh`.



Para este script nos logueamos en un terminal como root y ejecutamos:

```
[root@nodo1 ~]#  
/u01/app/oracle/product/11.1.0/db_1/root.sh  
Running Oracle 11g root.sh script...
```

```
The following environment variables are set as:  
ORACLE_OWNER= oracle  
ORACLE_HOME= /u01/app/oracle/product/10.2.0.4
```

Se nos pide que introduzcamos el directorio del bin local, en nuestro caso **/usr/bin**

```
Enter the full pathname of the local bin directory:  
[/usr/local/bin]: /usr/bin  
Copying dbhome to /usr/bin ...  
Copying oraenv to /usr/bin ...  
Copying coraenv to /usr/bin ...  
  
Creating /etc/oratab file...  
Entries will be added to the /etc/oratab file as  
needed by  
Database Configuration Assistant when a database is  
created  
Finished running generic part of root.sh script.  
Now product-specific root actions will be performed.  
Finished product-specific root actions.
```

Después de realizar esto podemos ya finalizar la instalación en la siguiente pantalla.



Instalación de ASM

Ahora vamos a acometer la instalación y configuración de ASM. Lo hacemos en este momento ya que el software de la base de datos 11g lo tenemos instalado y es necesario para todas las operaciones de ASM.

Antes, en el apartado de Premigración hemos instalado dos rpms, `oracleasm-support` y `oracleasm*el5-2*`, y la librería `oracleasmlib`, además de preparar el almacenamiento, requisitos indispensables para iniciar la instalación.

Empezamos configurando `$ORACLE_HOME` con el path que queremos para nuestra tarea mediante:

```
[oracle@nod01 ~]$ . oraenv
ORACLE_SID = [RAC1] ?
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/product/11.1.0/db_1/
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1/ is
/u01/app/oracle
```

Una vez hecho esto iniciamos el asistente de configuración de base de datos:

```
[oracle@nod01 ~]$ dbca
```

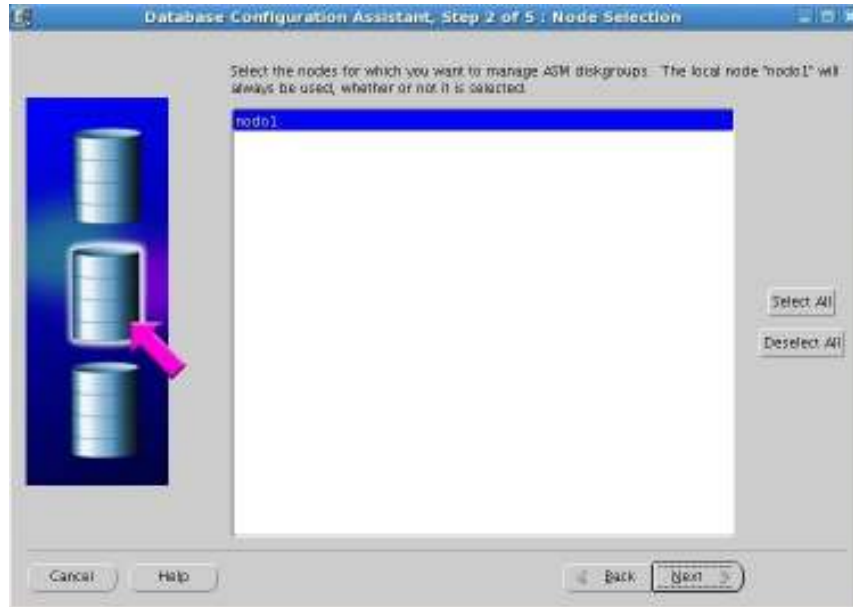
En la pantalla de inicio de dbca elegimos la opción **Oracle Real Application Clusters Database**:



En la segunda pantalla que nos aparece elegimos *Configure Automatic Storage Management*:



En la siguiente pantalla elegimos los nodos para los cuales deseamos administrar grupos de disco ASM. Seleccionamos el **node1**:



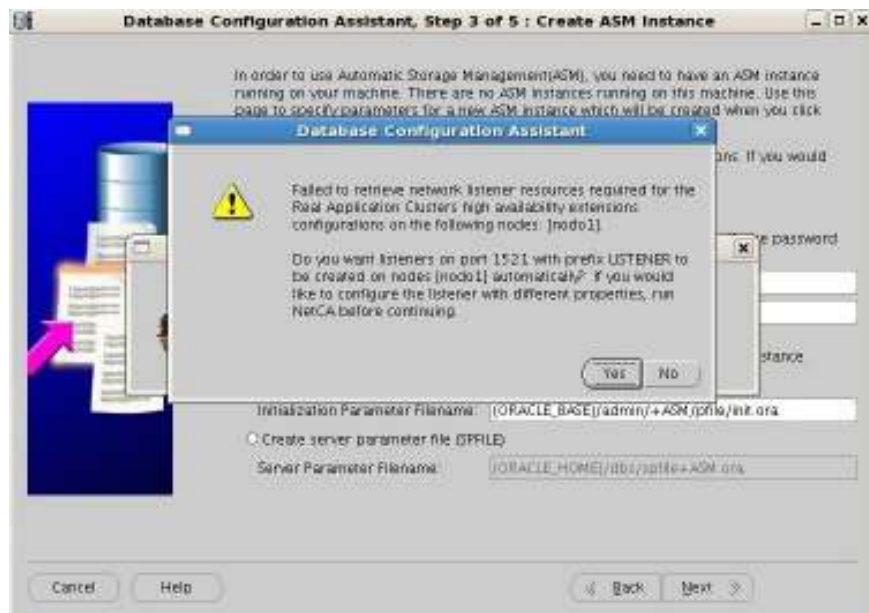
En la siguiente pantalla introducimos el **password para SYS** y seleccionamos la opción de **crear un archivo de parámetro de inicialización (IFILE)**:



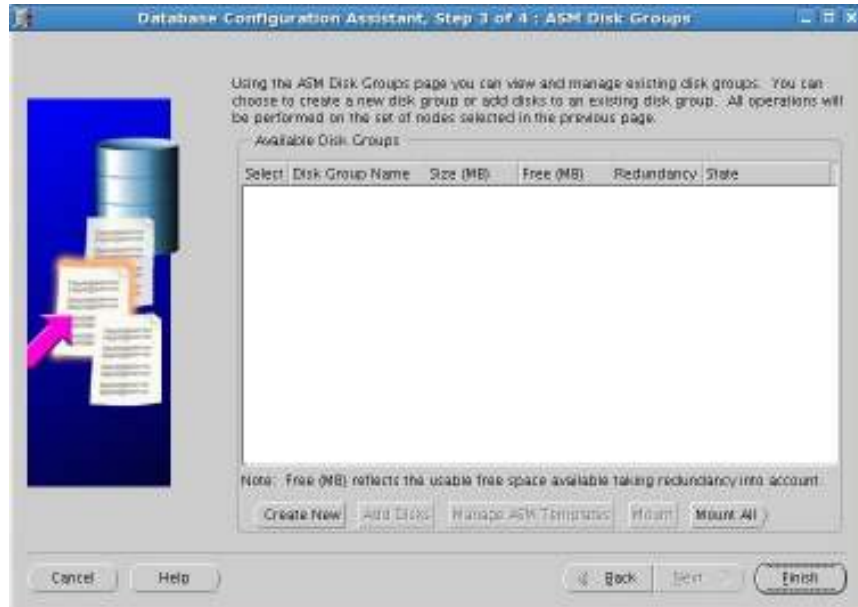
Al clicar en *Next* nos aparece una nueva ventana con un aviso en el que se nos indica que DBCA creará e iniciará la instancia ASM y que después de la inicialización podremos crear grupos de discos para el almacenamiento de la base de datos. Clicamos en **OK**:



Ahora aparece otra pantalla en la que nos pregunta si queremos que se creen los listeners por defecto. Aceptamos clicando en **Yes**:



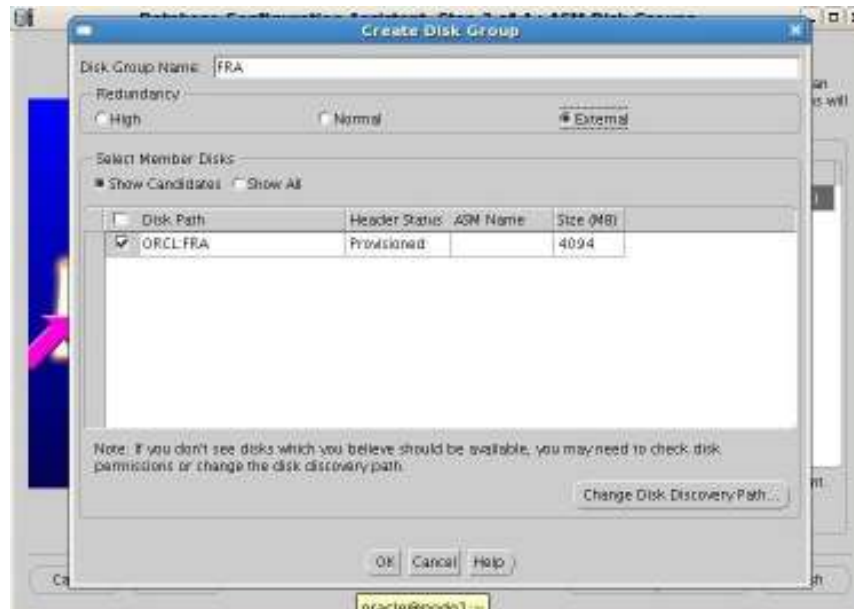
En la siguiente pantalla podemos realizar la configuración de los grupos de disco, **crear nuevos grupos** o añadir discos a los existentes, en nuestro caso, debemos crearlos. Para ello clicamos sobre *Create new*:



En la pantalla que nos aparece ahora podemos ver los discos que hemos configurado anteriormente, DATA y FRA. Seleccionamos DATA y Redundancia Externa y creamos el grupo con el nombre DATA:



Repetimos lo hecho para crear el grupo FRA:



La creación de los grupos de disco es el último paso de esta instalación, con lo que concluimos clicando en *Finish*:

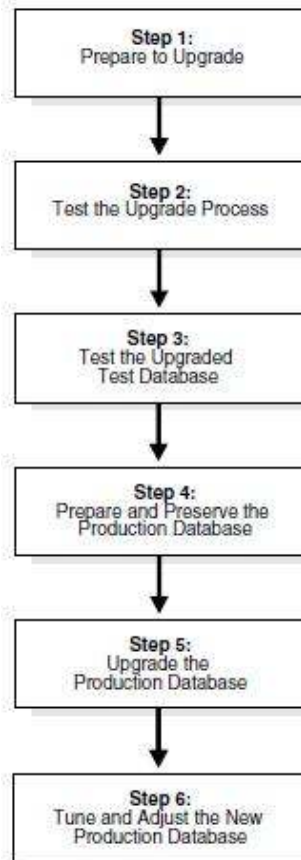


Ahora ya **tenemos configurada nuestra instancia ASM en cluster**, cuyo nombre aparece en el oratab como +ASM1.

Upgrade de single instance 10.2.0.4 a single instance 11.1.0.6

Una vez llegados aquí llega el momento de hacer la parte más importante del proyecto, la **migración**. La migración es un proceso costoso que consta de varios pasos que vamos a desgranar y a mostrar su ejecución en este apartado.

A continuación se muestra un gráfico con los pasos que se debe seguir en la migración de nuestra base de datos single instance 10.2.0.3 a single instance 11.1.0.6.



Paso 1 - Preparación del upgrade

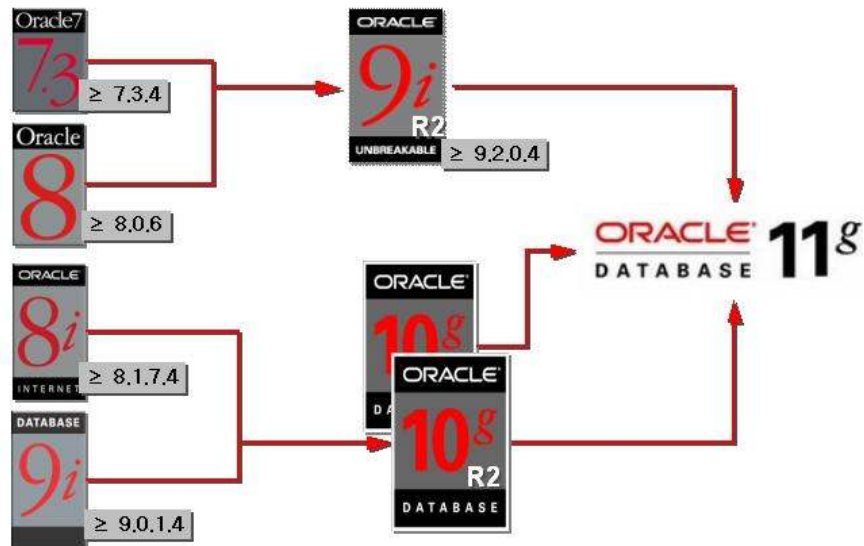
El objetivo fundamental del apartado es la determinación de la forma en la que se va a migrar la base de datos, la elección del método de upgrade a emplear y del nuevo home que se va a utilizar para la nueva versión.

La preparación del upgrade también consta de varias tareas propuestas para conseguir una mayor familiaridad con las nuevas características de la base de datos 11g, cosa que ya hemos realizado en el apartado **¿Por qué se migra?**

Hay que tener en cuenta que en realidad todas las **tareas necesarias de preparación del upgrade ya se han realizado** en los apartados explicados anteriormente, por lo tanto aquí sólo se hace una breve mención a unos puntos de interés.

Determinar el camino para el Upgrade

El camino a recorrer para realizar el upgrade a Oracle 11g depende del número de versión de la actual base de datos. Podría ser que no fuera posible el upgrade directo desde nuestra versión a 11g. Sin embargo, como podemos comprobar en el gráfico inferior, nuestra versión, una 10g R2, puede ser **directamente subida al nivel de 11g** sin pasos intermedios entre versiones.



Elección del método de upgrade

Se puede elegir diferentes métodos para llevar a cabo la migración. Las posibles elecciones son las siguientes:

- Database Upgrade Assistant
- Upgrade manual
- Export/Import

Por su simplicidad y fácil manejo elegiríamos Database Upgrade Assistant (dbua). Sin embargo, **elegimos Upgrade Manual** para mostrar de una forma más detallada todos los procesos y tareas que engloba la migración.

Elección del ORACLE_HOME

Se debe elegir un nuevo ORACLE_HOME para la nueva versión de base de datos que debe ser diferente del actual. No se puede instalar el nuevo software en el mismo ORACLE_HOME de la actual base de datos a no ser que estuviéramos instalando un patch set, que no es el caso.

Utilizando directorios de instalación separados conseguimos **mantener el software existente instalado** a la vez que añadimos el nuevo. Este método nos permite **probar el proceso de upgrade** en una base de datos de test antes de reemplazar todo el entorno de producción enteramente.

El nuevo ORACLE_HOME será `/u01/app/oracle/product/11.1.0/db_1`. Este directorio ya se ha creado en el apartado de preparación para la base de datos.

Paso 2 - Probar el proceso de upgrade

Se debe tener un entorno de test que sea igual o reducido respecto al de producción para, antes de realizar cualquier cambio en el entorno de producción y poder provocar problemas, probar cualquier operación en este entorno. En nuestro caso, **no se puede realizar** ya que no disponemos de todo lo necesario y realmente, nuestro

entorno imita a uno real de producción por lo que en sí, se lo podría considerar más un entorno de test.

Igualmente, el fin de este paso es que antes de realizar el upgrade en nuestro entorno de producción lo realizamos sobre nuestra base de datos de test para asegurar que el upgrade será realizado sin ningún problema y con unos resultados óptimos.

Paso 3 – Probar la base de datos de test actualizada

Se debe comprobar el funcionamiento de las aplicaciones que trabajan con la base de datos y verificar que operan de una forma adecuada. En el caso de que su funcionalidad no fuera la correcta se debe buscar la causa y realizarles también un upgrade o las modificaciones que fueran necesarias.

Como en el paso anterior, **este paso lo saltamos** ya que, en este caso, nuestro entorno ni siquiera consta de aplicaciones que probar por lo que no realizamos comprobaciones.

Paso 4 – Preparar y preservar la base de datos

Nos encontramos ya en el paso previo a la migración en sí de la base de datos. Como dice el título del paso, aquí vamos a realizar las últimas tareas para preparar la base de datos para la migración y para que la misma no afecte a los datos ni al funcionamiento de la base de datos. Este paso consta de tres tareas diferentes:

- Instalar el software de la base de datos 11.1.0.6
- Ejecutar Pre-Upgrade Information Tool
- Preparar la base de datos para realizarle el upgrade

Instalar el software de la base de datos 11.1.0.6

El software de la base de datos **ya está instalado**, como hemos mostrado en un apartado anterior.

Ejecutar Pre-Upgrade Information Tool

Una vez instalado ya el software de la base de datos se debe analizar nuestra base de datos actual antes de realizar el upgrade.

El análisis lo hacemos mediante **Pre-Upgrade Information Tool**. Este paso es recomendado aunque el upgrade se realice mediante DBUA. Como nosotros lo vamos a hacer todo manualmente lo realizamos.

Para ejecutar Pre-Upgrade Information Tool realizamos los siguientes pasos:

1. Nos logueamos como `oracle` y vamos al directorio home de la nueva base de datos `11g`.
2. Copiamos el Pre-Upgrade Information Tool (`ut1u111i.sql`) situado en `./rdbms/admin`, lo pegamos en el directorio temporal del sistema `/tmp` y nos situamos en el.
3. Arrancamos SQL*Plus y nos conectamos como `SYSDBA`:

```
sqlplus /nolog
SQL> connect / as sysdba
```

Connected.

4. Indicamos al sistema que direccione los resultados a un archivo de log para un análisis posterior:

```
SQL> SPOOL upgrade_info.log
```

5. Ejecutamos el Pre-Upgrade Information Tool:

```
SQL> @utlu111i.sql
```

6. Cerramos el direccionamiento al archivo de log:

```
SQL> SPOOL OFF
```

7. Comprobamos el **resultado de la ejecución del Pre-Upgrade Information Tool** en upgrade_info.log

```
Oracle Database 11.1 Pre-Upgrade Information Tool      08-20-2009 11:24:11
.
*****
Database:
*****
--> name:          DBASE10
--> version:       10.2.0.4.0
--> compatible:    10.2.0.1.0
--> blocksize:     8192
--> platform:      Linux IA (32-bit)
--> timezone file: V4
.
*****
Tablespaces: [make adjustments in the current environment]
*****
--> SYSTEM tablespace is adequate for the upgrade.
.... minimum required size: 740 MB
.... AUTOEXTEND additional space required: 240 MB
--> UNDOTBS1 tablespace is adequate for the upgrade.
.... minimum required size: 321 MB
--> SYSAUX tablespace is adequate for the upgrade.
.... minimum required size: 429 MB
.... AUTOEXTEND additional space required: 169 MB
--> TEMP tablespace is adequate for the upgrade.
.... minimum required size: 61 MB
.... AUTOEXTEND additional space required: 31 MB
--> EXAMPLE tablespace is adequate for the upgrade.
.... minimum required size: 69 MB
.
*****
Update Parameters: [Update Oracle Database 11.1 init.ora or spfile]
*****
WARNING: --> "sga_target" needs to be increased to at least 336 MB
.
*****
Renamed Parameters: [Update Oracle Database 11.1 init.ora or spfile]
*****
-- No renamed parameters found. No changes are required.
.
*****
Obsolete/Deprecated Parameters: [Update Oracle Database 11.1 init.ora or
spfile]
*****
--> "background_dump_dest" replaced by "diagnostic_dest"
--> "user_dump_dest" replaced by "diagnostic_dest"
--> "core_dump_dest" replaced by "diagnostic_dest"
.
*****
```

```
Components: [The following database components will be upgraded or
installed]
*****
--> Oracle Catalog Views           [upgrade]  VALID
--> Oracle Packages and Types      [upgrade]  VALID
--> JServer JAVA Virtual Machine   [upgrade]  VALID
--> Oracle XDK for Java            [upgrade]  VALID
--> Oracle Workspace Manager       [upgrade]  VALID
--> OLAP Analytic Workspace        [upgrade]  VALID
--> OLAP Catalog                   [upgrade]  VALID
--> EM Repository                  [upgrade]  VALID
--> Oracle Text                    [upgrade]  VALID
--> Oracle XML Database            [upgrade]  VALID
--> Oracle Java Packages           [upgrade]  VALID
--> Oracle interMedia              [upgrade]  VALID
--> Spatial                        [upgrade]  VALID
--> Data Mining                    [upgrade]  VALID
--> Expression Filter              [upgrade]  VALID
--> Rule Manager                   [upgrade]  VALID
--> Oracle OLAP API                [upgrade]  VALID
.
*****
Miscellaneous Warnings
```

```
WARNING: --> Database contains stale optimizer statistics.
.... Refer to the 11g Upgrade Guide for instructions to update
.... statistics prior to upgrading the database.
.... Component Schemas with stale statistics:
....   SYS
....   WMSYS
....   OLAPSYS
....   SYSMAN
....   CTXSYS
....   XDB
....   MDSYS
WARNING: --> Database contains schemas with objects dependent on network
packages.
.... Refer to the 11g Upgrade Guide for instructions to configure Network
ACLs.
WARNING: --> EM Database Control Repository exists in the database.
.... Direct downgrade of EM Database Control is not supported. Refer to
the 11g Upgrade Guide for instructions to save the EM data prior to
upgrade.
```

En el resultado de la ejecución del Pre-Upgrade Information Tool vemos que aparecen varios **errores y warnings**. Ahora vamos a mostrar el tratamiento que les damos:

- **WARNING: --> "sga_target" needs to be increased to at least 336 MB**

Este warning nos indica que el parámetro **sga_target** tiene un tamaño inadecuado para la nueva versión de la base de datos. Conviene reconfigurarlo para el correcto funcionamiento de la base de datos una vez hecha la migración.

Para la edición de estos parámetros debemos reescribir el fichero de arranque de la base de datos. Este fichero de arranque puede ser un **init.ora**, **spfile** o **pfile**. Desde la versión 9 de la base de datos Oracle se recomienda utilizar **spfile** ya que es binario, permite arrancar la base de datos sin estar en su mismo sistema de ficheros y con un ALTER SYSTEM se actualiza automáticamente, cosa que no pasa con los **init.ora**; entre otras características.

Nuestra base de datos actual utiliza **spfile**. Lo comprobamos y obtenemos su ubicación mediante:

```
SQL> show parameter spfile

NAME                                 TYPE
-----
VALUE
-----
spfile                               string
/u01/app/oracle/product/10.2.0/db_1/dbs/spfilebase.ora
```

Ahora debemos editarlo. Sin embargo, el fichero **spfile** es binario, por lo que no podemos hacerlo. Para hacerlo debemos crear un archivo **pfile**, que sí podemos editar, y hacer los cambios ahí. Además, el archivo **pfile** sólo se utiliza si no existe **spfile** y sólo cuando se arranca la base de datos (los **spfile** realizan cambios dinámicamente mientras la base de datos está levantada). **Creamos el pfile:**

```
SQL> create pfile from spfile;
File created.
```

El siguiente paso es copiar el archivo **pfile**, llamado **initDBase10.ora** (al no especificar un nombre en el comando lo crea con este), a la localización que debe tener en el home de la 11g, es decir, `/u01/app/oracle/product/11.1.0/db_1/dbs`. Lo editamos indicando un valor acorde a lo que necesita la nueva base de datos (igual o superior a 336 MB):

```
*.sga_target=340212672
```

- **Obsolete/Deprecated Parameters**

Lo siguiente que debemos configurar es un conjunto de parámetros (**background_dump_dest**, **user_dump_dest** y **core_dump_dest**) que en la nueva versión de la base de datos han sido sustituidos por uno solo, **diagnostic_dest**. Para ello debemos seguir los mismos pasos que en el punto anterior, editar el **pfile** que hemos creado.

Estos parámetros indican un destino de volcado para la memoria, **background** y **usuario**. Lo comprobamos:

```
SQL> show parameter background_dump_dest

NAME                                 TYPE
-----
```

```
VALUE
-----
background_dump_dest          string
/u01/app/oracle/diag/rdbms/dbase10/DBase10/t
race

SQL> show parameter user_dump_dest
NAME                           TYPE
-----
VALUE
-----
user_dump_dest                 string
/u01/app/oracle/diag/rdbms/dbase10/DBase10/t
race

SQL> show parameter core_dump_dest
NAME                           TYPE
-----
VALUE
-----
core_dump_dest                string
/u01/app/oracle/diag/rdbms/dbase10/DBase10/c
dump
```

En la nueva versión de la base de datos , como hemos comentado antes, son sustituidos por un único destino, `diagnostic_dest`. Hay dos opciones, elegir un nuevo destino creado por nosotros o simplemente utilizar el `$ORACLE_BASE`. Como esta es la opción recomendada, es la que utilizamos. Eliminamos los antiguos parámetros y añadimos `diagnostic_dest`, quedando así:

```
SQL> show parameter diagnostic_dest
NAME                           TYPE
-----
VALUE
-----
diagnostic_dest               string
/u01/app/oracle
```

- **WARNING: --> Database contains stale optimizer statistics.**

Este warning nos indica que **las estadísticas del optimizador no están actualizadas**. Cuando se hace un upgrade a 11g las estadísticas del optimizador son recogidas para las tablas del diccionario que no tienen estadísticas. Esta recolección de estadísticas puede llevar tiempo para bases de datos con un gran número de tablas de diccionario, pero la reunión sólo se realiza para aquellas tablas que carecen de estadísticas o son cambiadas significativamente durante el upgrade.

Para determinar los esquemas que carecen de estadísticas ejecutamos el script **check_stale_stats.sql**:

```
SQL>@ $HOME/migra/check_stale_stats.sql
```

El resultado lo mostramos a continuación:

```
-- There are no stale statistics in CTXSYS schema.
```

```
-- There are no stale statistics in DMSYS schema.

-- There are no stale statistics in EXFSYS schema.

-- There are no stale statistics in MDSYS schema.

-----
-----
-- OLAPSYS schema contains stale statistics use the following to gather
the statistics --
-----
-----
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS('OLAPSYS',OPTIONS=>'GATHER',
ESTIMATE_PERCENT => DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL
COLUMNS SIZE AUTO', CASCADE => TRUE);

-- There are no stale statistics in ORDSYS schema.

-----
-----
-- SYS schema contains stale statistics use the following to gather the
statistics --
-----
-----
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS('SYS',OPTIONS=>'GATHER',
ESTIMATE_PERCENT => DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL
COLUMNS SIZE AUTO', CASCADE => TRUE);

-----
-----
-- SYSMAN schema contains stale statistics use the following to gather
the statistics --
-----
-----
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS('SYSMAN',OPTIONS=>'GATHER',
ESTIMATE_PERCENT => DBMS_STATS.AUTO_SAMPLE_SIZE, METHOD_OPT => 'FOR ALL
COLUMNS SIZE AUTO', CASCADE => TRUE);

-- There are no stale statistics in WMSYS schema.

-- There are no stale statistics in XDB schema.
```

Comprobamos que los esquemas **OLAPSYS**, **SYS**, **SYSMAN** contienen estadísticas desactualizadas. Así pues, ejecutamos los procedimientos que se nos indica en el resultado mostrado (resaltados en negrita) para recoger las estadísticas. En todos los casos se devuelve:

PL/SQL procedure sucessfully completed

A continuación mostramos la lista de los componentes del sistema y los esquemas que son chequeados para la recolección de estadísticas durante el upgrade:

Nombre de Componente	Esquema
JServer JAVA Virtual Machine	SYS
OLAP Analytic Workspace	SYS
Oracle Database Catalog Views	SYS
Oracle Database JAVA Packages	SYS

Oracle Database Packages and Types	SYS, DBSNMP, OUTLN, SYSTEM, DIP
Oracle Database Vault	DVSYSD
Oracle Enterprise Manager	SYSMAN
Oracle Expression Filter	EXFSYS
Oracle Multimedia	ORDSYS, ORDPLUGINS, SI_INFORMTN _SCHEMA
Oracle Label Security	LBACSYS
Oracle OLAP API	SYS
Oracle Spatial	MDSYS, MDDATA
Oracle Text	CTXSYS
Oracle Ultra Search	WKSYS, WKPROXY, WK_ TEST
Oracle Workspace Manager	WMSYS
Oracle XDK	SYS
Oracle XML Database	XDB

- **WARNING: --> Database contains schemas with objects dependent on network packages.**

Este warning nos indica que hay esquemas con objetos dependiente de paquetes de red. Para comprobarlo ejecutamos en sqlplus:

```
SQL> SELECT * FROM DBA_DEPENDENCIES WHERE  
referenced_name IN ('UTL_TCP', 'UTL_SMTP',  
'UTL_MAIL', 'UTL_HTTP', 'UTL_NADDR') AND owner  
NOT IN ('SYS', 'PUBLIC', 'ORDPLUGINS');  
  
no rows selected
```

El resultado, como vemos es no rows selected esto nos indica que, en realidad, **no** hay dependencias.

Buscando el error en metalink.oracle.com encontramos que esto es debido al **bug 8333492**, en el cual se indica que la consulta SQL que se ejecuta para chequear las condiciones en el script utlu111i.sql (script de Pre-Upgrade Information Tool) hace referencia a 'ORD_PLUGINS' y el nombre correcto del esquema es 'ORDPLUGINS', esquema que hemos utilizado en nuestra consulta.

Como resultado de utilizar un nombre erróneo en la consulta se devuelve una row. Es por esto que **podemos obviar sin miedo este warning**. Otra solución es editar el script y cambiar el nombre del schema.

- **WARNING: --> EM Database Control Repository exists in the database.**

Este warning nos indica que debemos salvar los datos y archivos de control del Enterprise Manager de la base de datos antes de realizar la migración. Para ello vamos a utilizar la herramienta **emdwrdr**.

emdwgrd reside en el \$ORACLE_HOME/bin de la base de datos 11g y para utilizarla debemos configurar las variables de entorno PATH, LD_LIBRARY_PATH y SHLIB_PATH para que apunten al home de la nueva base de datos (/u01/app/oracle/product/11.1.0/db_1/):

```
[oracle@nodol bin]$ export PATH=$PATH:  
/u01/app/oracle/product/11.1.0/db_1  
  
[oracle@nodol bin]$ export  
LD_LIBRARY_PATH=/u01/app/oracle/product/11.1  
.0/db_1  
  
[oracle@nodol bin]$ export  
SHLIB_PATH=/u01/app/oracle/product/11.1.0/db  
_1
```

Antes de la ejecución de la herramienta debemos crear un directorio donde vuelque el resultado de la ejecución, es decir, el backup de estos datos y ficheros. Al directorio lo llamaremos backups y lo localizamos en \$HOME/setups/:

```
[oracle@nodol setups]$ mkdir backups
```

Una vez hecho todo esto ejecutamos **emdwgrd** de la siguiente forma:

```
[oracle@nodol bin]$ ./emdwgrd -save -sid  
DBase10 -path /home/oracle/setups/backups
```

Se nos pide que introduzcamos el password para el usuario SYS:

```
Enter sys password for database DBase10?
```

La ejecución nos muestra lo siguiente:

```
Wed Sep 23 20:18:59 2009 - Verify EM DB Control  
files ... pass  
Wed Sep 23 20:18:59 2009 - Validating DB  
Connection to DBase10 ... pass  
ENV var EM_REMCP not defined, check if rcp or  
scp is configured.  
Wed Sep 23 20:19:13 2009 - Creating directory  
... created  
Wed Sep 23 20:19:15 2009 - Stopping DB Control  
... stopped  
Wed Sep 23 20:19:38 2009 - Saving DB Control  
files  
... saved  
Wed Sep 23 20:20:14 2009 - Recompiling invalid  
objects ... recompiled  
Wed Sep 23 20:20:51 2009 - Exporting sysman  
schema for DBase10 ... exported  
Wed Sep 23 20:25:44 2009 - DB Control was saved  
successfully.  
Wed Sep 23 20:25:44 2009 - Starting DB Control  
... started  
Wed Sep 23 20:27:54 2009 - Dump directory was  
dropped successfully.
```

Como se puede comprobar en el resultado mostrado, **todas las operaciones se realizan con éxito** con lo que podemos dar por finalizado el proceso.

Preparar la base de datos para realizarle el upgrade

En este apartado lo primero que vamos realizar es un **chequeo de la integridad de la base de datos** que vamos a migrar. Para realizar esto

ejecutaremos el script **dbupgdiag.sql** antes de iniciar el upgrade. Para ello nos situamos en el directorio `$HOME/migra` y ejecutamos:

```
[oracle@nodol1 migra]$ sqlplus / as sysdba
SQL> @dbupgdiag.sql
```

Seguidamente, nos aparece un mensaje pidiendo un directorio donde volcar los resultados. Le indicamos `/tmp`:

```
Enter location for Spooled output as
Parameter1:
Enter value for 1: /tmp
```

El resultado de esta ejecución se guarda donde le hemos indicado en el fichero **db_upg_diag_Dbase10_22-Sep-2009_0704.1og**. El resultado es el siguiente:

```
*** Start of LogFile ***

Oracle Database Upgrade Diagnostic Utility      09-22-2009 19:04:20

=====
Database Uptime
=====

17:08 22-SEP-09

=====
Database Wordsize
=====

This is a 32-bit database

=====
Software Verison
=====

Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - Prod
PL/SQL Release 10.2.0.4.0 - Production
CORE    10.2.0.4.0      Production
TNS for Linux: Version 10.2.0.4.0 - Production
NLSRTL Version 10.2.0.4.0 - Production

=====
Compatibility
=====

Compatibility is set as 10.2.0.1.0

=====
Component Status
=====

Comp ID Component                               Status    Version    Org_Version  Prv_Version
-----
AMD      OLAP Catalog                                  VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
APS      OLAP Analytic Workspace                      VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
CATALOG  Oracle Database Catalog Views                VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
CATJAVA  Oracle Database Java Packages                VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
CATPROC  Oracle Database Packages and Types           VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
CONTEXT  Oracle Text                                  VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
EM       Oracle Enterprise Manager                    VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
EXF      Oracle Expression Filter                      VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
JAVAVM   JServer JAVA Virtual Machine                 VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
ODM      Oracle Data Mining                            VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
ORDIM    Oracle interMedia                             VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
OWM      Oracle Workspace Manager                     VALID    10.2.0.4.3  10.2.0.1.0   10.2.0.1.0
RUL      Oracle Rule Manager                           VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
SDO      Spatial                                       VALID    10.2.0.4.0  10.2.0.1.0   10.2.0.1.0
```

XDB	Oracle XML Database	VALID	10.2.0.4.0	10.2.0.1.0	10.2.0.1.0
XML	Oracle XDK	VALID	10.2.0.4.0	10.2.0.1.0	10.2.0.1.0
XOQ	Oracle OLAP API	VALID	10.2.0.4.0	10.2.0.1.0	10.2.0.1.0

=====
List of Invalid Database Objects Owned by SYS / SYSTEM
=====

Number of Invalid Objects

There are no Invalid Objects

DOC>#####
DOC>
DOC> If there are no Invalid objects below will result in zero rows.
DOC>
DOC>#####
DOC>#

no rows selected

=====
List of Invalid Database Objects
=====

Number of Invalid Objects

There are no Invalid Objects

DOC>#####
DOC>
DOC> If there are no Invalid objects below will result in zero rows.
DOC>
DOC>#####
DOC>#

no rows selected

=====
Identifying whether a database was created as 32-bit or 64-bit
=====

DOC>#####
DOC>
DOC> Result referencing the string 'B023' ==> Database was created as 32-bit
DOC> Result referencing the string 'B047' ==> Database was created as 64-bit
DOC> When String results in 'B023' and when upgrading database to 10.2.0.3.0
DOC> (64-bit) , For known issue refer below articles
DOC>
DOC> Note 412271.1 ORA-600 [22635] and ORA-600 [KOKEIIX1] Reported While
DOC> Upgrading Or Patching Databases To 10.2.0.3
DOC> Note 579523.1 ORA-600 [22635], ORA-600 [KOKEIIX1], ORA-7445 [KOPESIZ] and
DOC> OCI-21500 [KOXSIHREAD1] Reported While Upgrading To 11.1.0.6
DOC>
DOC>#####
DOC>#

Metadata Initial DB Creation Info

B023 Database was created as 32-bit

=====
Number of Duplicate Objects Owned by SYS and SYSTEM
=====

Counting duplicate objects

COUNT(1)

4

=====

Duplicate Objects Owned by SYS and SYSTEM
=====

Querying duplicate objects

OBJECT_NAME	OBJECT_TYPE
-----	-----
AQ\$_SCHEDULES	TABLE
AQ\$_SCHEDULES_PRIMARY	INDEX
DBMS_REPCAT_AUTH	PACKAGE
DBMS_REPCAT_AUTH	PACKAGE BODY

DOC>
DOC>#####
DOC>
DOC> If any objects found please follow below article.
DOC> Note 1030426.6 How to Clean Up Duplicate Objects Owned by SYS and SYSTEM schema
DOC> Read the Exceptions carefully before taking actions.
DOC>
DOC>#####
DOC>#

=====
JVM Verification
=====

JAVAVM - Installed properly

=====
Checking Existence of Java-Based Users and Roles
=====

DOC>
DOC>#####
DOC>
DOC> There should not be any Java Based users for database version 9.0.1 and above.
DOC> If any users found, it is faulty JVM.
DOC>
DOC>#####
DOC>#

User Existence

No Java Based Users

DOC>
DOC>#####
DOC>
DOC> Healthy JVM Should contain Six Roles.
DOC> If there are more or less than six role, JVM is inconsistent.
DOC>
DOC>#####
DOC>#

Role

There are 6 JAVA related roles

Roles

ROLE

JAVAUSERPRIV
JAVAIDPRIV
JAVASYSPRIV
JAVADEBUGPRIV
JAVA_ADMIN
JAVA_DEPLOY

=====
List of Invalid Java Objects owned by SYS
=====

```
There are no SYS owned invalid JAVA objects

DOC>
DOC>#####
DOC>
DOC> Check the status of the main JVM interface packages DBMS_JAVA
DOC> and INITJVMAUX and make sure it is VALID.
DOC> If there are no Invalid objects below will result in zero rows.
DOC>
DOC>#####
DOC>#

no rows selected

INFO: Below query should succeed with 'foo' as result.

JAVAVM TESTING
-----
foo

*** End of LogFile ***
```

Como se puede observar en los resultados expuestos, **no hay objetos inválidos**. Sin embargo se recomienda la ejecución del script **utlrp.sql**, el cual realiza una validación de todos los objetos inválidos de la base de datos. Para ello ejecutamos:

```
SQL> @ $ORACLE_HOME/rdbms/admin/utlrp.sql
```

El resultado de esta ejecución es:

```
TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_BGN 2009-09-22 19:22:06

DOC> The following PL/SQL block invokes UTL_RECOMP to recompile invalid
DOC> objects in the database. Recomilation time is proportional to the
DOC> number of invalid objects in the database, so this command may take
DOC> a long time to execute on a database with a large number of invalid
DOC> objects.
DOC>
DOC> Use the following queries to track recompilation progress:
DOC>
DOC> 1. Query returning the number of invalid objects remaining. This
DOC> number should decrease with time.
DOC> SELECT COUNT(*) FROM obj$ WHERE status IN (4, 5, 6);
DOC>
DOC> 2. Query returning the number of objects compiled so far. This number
DOC> should increase with time.
DOC> SELECT COUNT(*) FROM UTL_RECOMP_COMPILED;
DOC>
DOC> This script automatically chooses serial or parallel recompilation
DOC> based on the number of CPUs available (parameter cpu_count) multiplied
DOC> by the number of threads per CPU (parameter parallel_threads_per_cpu).
DOC> On RAC, this number is added across all RAC nodes.
DOC>
DOC> UTL_RECOMP uses DBMS_SCHEDULER to create jobs for parallel
DOC> recompilation. Jobs are created without instance affinity so that they
DOC> can migrate across RAC nodes. Use the following queries to verify
DOC> whether UTL_RECOMP jobs are being created and run correctly:
```

```
DOC>
DOC> 1. Query showing jobs created by UTL_RECOMP
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>      WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#
```

PL/SQL procedure successfully completed.

TIMESTAMP

```
-----
COMP_TIMESTAMP UTLRP_END 2009-09-22 19:22:16
```

PL/SQL procedure successfully completed.

```
DOC> The following query reports the number of objects that have compiled
DOC> with errors (objects that compile with errors have status set to 3 in
DOC> obj$). If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#
```

OBJECTS WITH ERRORS

```
-----
0
```

```
DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#
```

ERRORS DURING RECOMPILATION

```
-----
0
```

PL/SQL procedure successfully completed.

Comprobamos en el resultado de la ejecución de utlrp.sql que no hay ningún objeto inválido, por lo que podemos continuar con los siguientes pasos.

Ahora vamos a comprobar que el tipo de datos **TIMESTAMP WITH TIMEZONE** es correcto. Esto se debe hacer ya que los archivos de zonas de tiempo que vienen con la base de datos 11g tienen la versión 4 para reflejar los cambios en las reglas de transición para algunas regiones y algunas versiones anteriores tienen, o pueden tener, la versión 2. Estos cambios pueden afectar a datos existentes con este tipo de datos.

Para comprobar nuestra versión realizamos la siguiente ejecución:

```
SQL> select * from v$timezone_file;
```

```
FILENAME          VERSION
```

timezlrq.dat 4

Como indica la consulta, la versión actual es la 4 con lo que no necesitamos hacer más operaciones sobre este asunto.

Ahora realizamos una **comprobación del diccionario** de la base de datos en busca de posibles corrupciones. Para ello ejecutamos el siguiente conjunto de sentencias en sqlplus:

```
SQL> Set verify off
SQL> Set space 0
SQL> Set line 120
SQL> Set heading off
SQL> Set feedback off
SQL> Set pages 1000
SQL> Spool analyze.sql
```

Y seguidamente ejecutamos la consulta siguiente:

```
SQL> SELECT 'Analyze cluster "'
||cluster_name||'" validate structure
cascade;' FROM dba_clusters WHERE
owner='SYS' UNION SELECT 'Analyze table
"'||table_name||'" validate structure
cascade;' FROM dba_tables WHERE owner='SYS'
AND partitioned='NO' AND (iot_type='IOT' OR
iot_type is NULL) UNION SELECT 'Analyze
table "'||table_name||'" validate structure
cascade into invalid rows;' FROM dba_tables
WHERE owner='SYS' AND partitioned='YES';

SQL> Spool off
```

El resultado de estas ejecuciones es la creación de un nuevo script llamado **analyze.sql**, tal y como hemos indicado con SQL> Spool analyze.sql antes de ejecutar. Este script analiza los objetos que forman el diccionario de nuestra base de datos. Este script es muy largo (aproximadamente 700 líneas) por lo que, a modo de muestra, seleccionamos lo siguiente:

```
Analyze cluster "C_COBJ#" validate structure
cascade;
Analyze cluster "C_FILE#_BLOCK#" validate
structure cascade;
Analyze cluster "C_MLOG#" validate structure
cascade;
Analyze cluster "C_USER#" validate structure
cascade;
Analyze cluster "SMON_SCN_TO_TIME" validate
structure cascade;
Analyze table "ACCESS$" validate structure
cascade;
Analyze table "ALERT_QT" validate structure
cascade;
Analyze table "APPLY$_CONF_HDLR_COLUMNS"
validate structure cascade;
```

Una vez creado el script pasamos a ejecutarlo. Para hacer esto necesitamos ejecutar primero el script **utlvalid.sql**. Este script tiene como función **crear la tabla por defecto para almacenar la salida** que produce el comando **validate** de **analyze** en tablas particionadas. Así pues, ejecutamos:

```
SQL> @$ORACLE_HOME/rdbms/admin/utlvalid.sql
SQL> @analyze.sql
```

Como resultado nos aparece lo siguiente:

```
SP2-0734: unknown command beginning "SQL>
SELEC..." - rest of line ignored.
SP2-0734: unknown command beginning "SQL>
spool..." - rest of line ignored.
```

Estos errores se pueden obviar ya que son producidos por la sentencia `SELECT` que hay al principio del script y por el `spool off` que hay al final, que han quedado metidos dentro cuando lo hemos creado.

Por último en este apartado vamos a **asegurarnos de que todos los refrescos de snapshot han sido completados con éxito y de que su replicación ha sido parada**. Para ello ejecutamos la siguiente sentencia:

```
SQL> SELECT distinct(trunc(last_refresh))
      FROM dba_snapshot_refresh_times;

(TRUNC(LA
-----
26-SEP-09
```

Esto significa que el único refresco que ha habido **se ha realizado con éxito** y que se produjo el 26 de septiembre de 2009.

Con esto concluimos este apartado.

Paso 5 - Hacer el Upgrade de la base de datos

Llegados ya a este punto vamos a realizar la migración de la base de datos propiamente dicha. Todos los pasos previos que hemos venido realizando han sido de cara a poder realizar este de la formas más completa, eficiente y segura posible.

Hay que tener en cuenta que gran parte del proceso que vamos a describir a continuación se debe realizar con la base de datos parada, con lo cual es crucial realizar todas las tareas perfectamente y lo más rápido posible ya que, en sistemas de producción críticos el sistema no puede estar parado más de un determinado tiempo.

El primer paso crítico es la **parada del listener** de la base de datos. Debemos parar el listener y eliminarlo para después crear uno nuevo para la base de datos actualizada. Esto lo vamos a hacer de la siguiente manera. Primero paramos el listener:

```
[oracle@nod01 ~]$ lsnrctl stop
```

Ahora ejecutamos el Network Configuration Assistant de nuestra base de datos actual para **eliminar el listener**:

```
[oracle@nod01 ~]$ netca
```

En la pantalla de inicio de Network Configuration Assistant elegimos la opción *Configuración del Listener*:



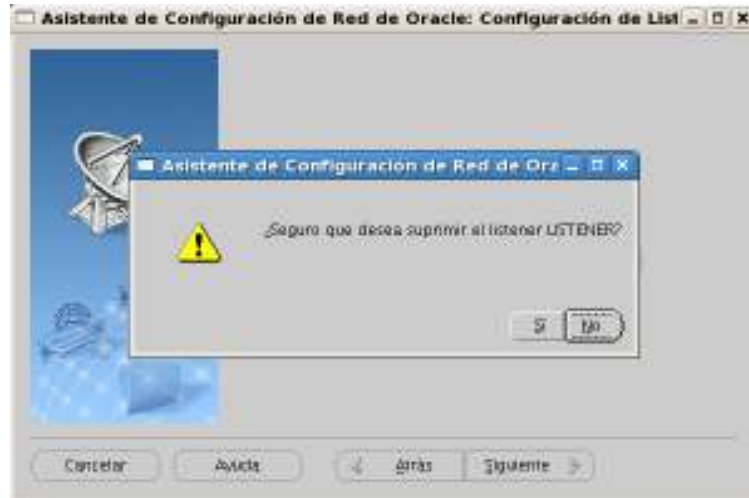
En la siguiente pantalla seleccionamos *Suprimir*:



Ahora seleccionamos el nombre de nuestro listener, LISTENER:



Al clicar sobre siguiente nos aparece un mensaje pidiendo la confirmación. Confirmamos:



El listener es eliminado con éxito:



Y por último ejecutamos el Network Configuration Assistant de la nueva versión de la base de datos para **crear el nuevo listener**. En este caso debemos desplazarnos hasta el directorio `bin` del home de la nueva base de datos para ejecutarlo directamente allí ya que aún no lo tenemos añadido al `$PATH`:

```
[oracle@nodo1 ~]$  
/u01/app/oracle/product/11.1.0/db_1/bin/netca
```

En este caso, en la pantalla inicial de Network Configuration Assistant elegimos la opción *Configuración del Listener*:



En la siguiente pantalla sólo nos aparece la opción de *Agregar* un listener ya que no hay ninguno definido en la máquina actualmente. Seleccionamos esta opción:



Ahora elegimos el nombre del nuevo listener. **Le llamamos LISTENER**, al igual que antes:



Elegimos de entre los protocolos disponibles **TCP** ya que va a ser el que utilizemos con la base de datos:



El último paso es elegir el puerto en el que escuchará LISTENER, el puerto por defecto es el 1521, que además es el que teníamos configurado hasta ahora en la base de datos. Así pues, **elegimos el 1521**:



Ahora nos aparece el mensaje de si queremos realizar otra configuración de listener. Seleccionamos *No*:



Por último se nos muestra el mensaje de éxito en la configuración del listener:



Una vez finalizado el trabajo referente a los listeners pasamos a realizar la siguiente tarea. Esta operación es la **parada de isqlplus y del Enterprise Manager**.

Para el caso de **isqlplus** ejecutamos:

```
[oracle@nod01 bin]$ isqlplusctl stop
iSQL*Plus 10.2.0.4.0
Copyright (c) 2003, 2007, Oracle. All Rights Reserved.
Stopping iSQL*Plus ...
iSQL*Plus stopped.
```

Ahora paramos **Enterprise Manager**, es decir, el dbconsole:

```
[oracle@nod01 bin]$ emctl stop dbconsole
TZ set to Europe/Madrid
Oracle Enterprise Manager 10g Database Control
Release 10.2.0.4.0
Copyright (c) 1996, 2007 Oracle Corporation. All rights reserved.
http://nod01.acshost.com:5500/em/console/aboutApplication
Stopping Oracle Enterprise Manager 10g Database Control ...
... Stopped.
```

Ya parados Enterprise Manager y isqlplus pasamos a realizar una **consulta para asegurarnos de que no hay archivos que necesiten de media recovery**:

```
SQL> select * from v$recover_file;
no rows selected
```

Evidentemente, al no devolver ninguna consulta indica que no hay ningún archivo que necesite de *media recovery* y, por lo tanto, podemos continuar.

Ahora comprobamos que **no hay archivos en modo backup**:

```
SQL> select * from v$backup where
status='NOT_ACTIVE';
```

FILE#	STATUS	CHANGE#	TIME
1	NOT ACTIVE	0	
2	NOT ACTIVE	0	
3	NOT ACTIVE	0	
4	NOT ACTIVE	0	

5 NOT ACTIVE

0

Mediante la siguiente consulta comprobamos que **no haya ninguna transacción distribuida pendiente de resolución**:

```
SQL> select * from dba_2pc_pending;
no rows selected
```

Al igual que en casos anteriores, no devuelve ninguna fila, por lo tanto no es necesario ninguna acción por nuestra parte y podemos pasar a comprobar que tanto el usuario **SYS** como **SYSTEM** tienen como **tablespace por defecto SYSTEM**:

```
SQL> select username, default_tablespace from
       dba_users where username in ('SYS','SYSTEM');
USERNAME                                DEFAULT_TABLESPACE
-----                                -
SYSTEM                                  SYSTEM
SYS                                      SYSTEM
```

Comprobamos que **aud\$ se encuentra en el tablespace SYSTEM**:

```
SQL> select tablespace_name from dba_tables where
       table_name='AUD$';
TABLESPACE_NAME
-----
SYSTEM
```

Comprobamos que la base de datos no tiene **ningún usuario SSL registrado**:

```
SQL> select name from sys.user$ where ext_username IS
       NOT NULL and password='GLOBAL';
no rows selected
```

A continuación vamos a realizar una **copia de seguridad de los ficheros de configuración** de la base de datos, listener.ora, tnsnames.ora, initDBase10.ora y oratab:

```
[oracle@nod01 ~]$ cp
  $ORACLE_HOME/network/admin/listener.ora
  /home/oracle/setups/backups/database
[oracle@nod01 ~]$ cp
  $ORACLE_HOME/network/admin/tnsnames.ora
  /home/oracle/setups/backups/database
[oracle@nod01 ~]$ cp $ORACLE_HOME/dbs/initDBase10.ora
  /home/oracle/setups/backups/database
[oracle@nod01 ~]$ cp /etc/oratab
  /home/oracle/setups/backups/database
```

Tras realizar esto último pasamos a tomar nota de la **localización de todos los datafiles, redologs y controlfiles**. Esto lo hacemos mediante:

```
SQL> SELECT name FROM v$controlfile;
NAME
-----
/u01/app/oracle/oradata/DBase10/control01.ctl
/u01/app/oracle/oradata/DBase10/control02.ctl
/u01/app/oracle/oradata/DBase10/control03.ctl

SQL> SELECT file_name FROM dba_data_files;
FILE_NAME
-----
```

```
/u01/app/oracle/oradata/DBase10/users01.dbf  
/u01/app/oracle/oradata/DBase10/sysaux01.dbf  
/u01/app/oracle/oradata/DBase10/undotbs01.dbf  
/u01/app/oracle/oradata/DBase10/system01.dbf  
/u01/app/oracle/oradata/DBase10/example01.dbf
```

```
SQL> SELECT group#, member FROM v$logfile;
```

```
GROUP#
```

```
-----
```

```
MEMBER
```

```
-----
```

```
          3  
/u01/app/oracle/oradata/DBase10/redo03.log  
          2  
/u01/app/oracle/oradata/DBase10/redo02.log  
          1  
/u01/app/oracle/oradata/DBase10/redo01.log
```

Después de tomar nota de lo anterior debemos realizar un **backup de la base de datos**. En este punto llegamos a **uno de los momentos más críticos de todo el proceso** ya que ahora **debemos parar por primera vez la base de datos**.

Para hacer el backup utilizaremos otra de las herramientas de Oracle, **RMAN**. Para utilizarlo debemos conectarnos a su interfaz mediante:

```
[oracle@nod01 bin]$ rman TARGET / NOCATALOG
```

Debemos ejecutar un proceso que haga todo el backup. Para ello asignamos un canal (channel) el cual, conecta con la base de datos y realiza las operaciones que le indicamos; hacer backup de la base de datos y de su controlfile:

```
run {  
  allocate channel c1 type disk;  
  backup database format  
    '/home/oracle/setups/backups/database/%U'TAG  
    before_upgrade;  
  backup current controlfile to  
    '/home/oracle/setups/backups/database/controlfile';}
```

Esta operación produce la siguiente salida por pantalla:

```
allocated channel: c1  
channel c1: sid=147 devtype=DISK  
  
Starting backup at 12-OCT-09  
channel c1: starting full datafile backupset  
channel c1: specifying datafile(s) in backupset  
input datafile fno=00001  
  name=/u01/app/oracle/oradata/DBase10/system01.dbf  
input datafile fno=00002  
  name=/u01/app/oracle/oradata/DBase10/undotbs01.dbf  
input datafile fno=00003  
  name=/u01/app/oracle/oradata/DBase10/sysaux01.dbf  
input datafile fno=00005  
  name=/u01/app/oracle/oradata/DBase10/example01.dbf  
input datafile fno=00004  
  name=/u01/app/oracle/oradata/DBase10/users01.dbf  
channel c1: starting piece 1 at 12-OCT-09  
channel c1: finished piece 1 at 12-OCT-09  
piece  
  handle=/home/oracle/setups/backups/database/0akrk9om  
  _1_1 tag=BEFORE_UPGRADE comment=NONE
```

```
channel c1: backup set complete, elapsed time:
 00:01:35
channel c1: starting full datafile backupset
channel c1: specifying datafile(s) in backupset
including current control file in backupset
including current SPFILE in backupset
channel c1: starting piece 1 at 12-OCT-09
channel c1: finished piece 1 at 12-OCT-09
piece
  handle=/home/oracle/setups/backups/database/0bkrk9rl
_1_1 tag=BEFORE_UPGRADE comment=NONE
channel c1: backup set complete, elapsed time:
 00:00:06
Finished backup at 12-OCT-09

Starting backup at 12-OCT-09
channel c1: starting full datafile backupset
channel c1: specifying datafile(s) in backupset
including current control file in backupset
channel c1: starting piece 1 at 12-OCT-09
channel c1: finished piece 1 at 12-OCT-09
piece
  handle=/home/oracle/setups/backups/database/controlf
ile tag=TAG20091012T144100 comment=NONE
channel c1: backup set complete, elapsed time:
 00:00:03
Finished backup at 12-OCT-09
released channel: c1
```

Como se puede comprobar en la salida mostrada, **no hay ningún mensaje de error**. Además comprobamos que en el directorio donde hemos indicado que se genere los archivos de backup, /home/oracle/setups/backups/database, **se ha generado el controlfile y dos nuevos archivos**:

```
[oracle@nod01 database]$ ls
0akrk9om_1_1 controlfile listener.ora
tnsnames.ora 0bkrk9rl_1_1 initDBase10.ora oratab
```

Ahora **desactivamos el archive log mode** ya que es preferible, antes de ejecutar la migración definitivamente, parar la recolección de los redologs ya que esto reducirá considerablemente el tiempo necesario para realizar la migración de la base de datos. Después de la migración lo volveremos a iniciar.

Para desactivar el archive log mode ejecutamos los siguientes comandos en sqlplus, logueados como sysdba:

```
SQL> shutdown immediate
SQL> startup mount
SQL> alter database noarchivelog;
SQL> alter database open;
SQL> alter system set log_archive_start=false
scope=spfile;
```

Ahora debemos cambiar la variable de entorno **\$ORACLE_HOME** y asegurarnos de que **\$ORACLE_BASE** es correcta y de que **\$PATH** contiene el directorio bin de la nueva versión de la base de datos.

Para cambiar la variable **\$ORACLE_HOME** **editamos el .bash_profile** de nuestro usuario, oracle, y la establecemos apuntando al nuevo home de la base de datos. Las variables para ORACLE en el .bash_profile quedan como sigue:

```
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=$ORACLE_BASE/product/11.1.0/db_1
PATH=$PATH:$HOME/bin:/sbin:/usr/sbin:$ORACLE_HOME/bin
ORACLE_SID=DBase10
```


Ahora, para hacer activos los cambios reejecutamos el archivo:

```
[oracle@nod01 bin]$ . .bash_profile
```

Comprobamos las variables:

```
[oracle@nod01 ~]$ echo $ORACLE_HOME
/u01/app/oracle/product/11.1.0/db_1
[oracle@nod01 ~]$ echo $ORACLE_BASE
/u01/app/oracle
[oracle@nod01 ~]$ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/usr/X
11R6/bin:/usr/java/jre1.5.0_10/bin:/home/oracle/bin:
/sbin:/usr/sbin:/u01/app/oracle/product/11.1.0/db_1/
bin:/usr/java/jre1.5.0_10/bin:/usr/java/jre1.5.0_10/
bin
```

Como se puede observar **todos los cambios se han realizado con éxito**.

Tras esto, **editamos el archivo oratab**. Ahora mismo el archivo apunta a nuestra base de datos actual con la siguiente entrada:

```
DBase10:/u01/app/oracle/product/10.2.0.4:Y
```

Cambiamos la entrada para que apunte al nuevo ORACLE_HOME y desactivamos el autoarranque cambiando la Y final por una N:

```
DBase10:/u01/app/oracle/product/11.1.0/db_1:N
```

Relacionado con esto ejecutamos por último **oraenv** para establecer el ORACLE_SID y que nos devuelva el ORACLE_BASE definitivo:

```
[oracle@nod01 ~]$ . oraenv
ORACLE_SID = [Dbase10] ? Dbase10
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.1.0/db_1 is
/u01/app/oracle
```

Y ahora, por fin, llega el paso más importante y crítico de la migración: la **ejecución del upgrade**.

Recordemos que la base de datos está parada en este momento y para ejecutar el upgrade necesitamos levantarla en el **modo upgrade**. Para ello nos desplazamos al directorio \$ORACLE_HOME/rdbms/admin y ejecutamos:

```
[oracle@nod01 admin]$ sqlplus /nolog
SQL> connect / as sysdba
SQL> startup upgrade
```

Ahora indicamos que la ejecución de todo lo que hagamos a partir de este momento sea guardado en el archivo upgrade.log:

```
SQL> set echo on
SQL> SPOOL upgrade.log
```

El siguiente paso es la **ejecución del script catupgrd.sql**, el cual determina qué scripts son necesarios para realizar el upgrade, los ejecuta y finalmente para la base de datos. Este script crea y realiza cambios en ciertas tablas del diccionario de datos.

La lista completa de todos los componentes que catupgrd.sql instala o actualiza a la nueva versión es:

- Oracle Database Catalog Views
- Oracle Database Packages and Types
- JServer JAVA Virtual Machine

- Oracle Database Java Packages
- Oracle XDK
- Oracle Real Application Clusters
- Oracle Workspace Manager
- Oracle Multimedia
- Oracle XML Database
- OLAP Analytic Workspace
- Oracle OLAP API
- OLAP Catalog
- Oracle Text
- Spatial
- Oracle Data Mining
- Oracle Label Security
- Messaging Gateway
- Oracle Expression Filter
- Oracle Rules Manager
- Oracle Enterprise Manager Repository
- Oracle Database Vault
- Oracle Application Express

Lo ejecutamos de la siguiente manera:

```
SQL> @catupgrd.sql
```

Como hemos indicado antes, toda la información generada es registrada en el archivo **upgrade.log**, el cual, tras toda la ejecución alcanza las 2026010 líneas. Es por esto que no lo mostramos aquí. Sólo apuntamos las últimas líneas para demostrar que se ejecutó en su totalidad y que, como hemos indicado antes, para la base de datos en el último paso de su ejecución:

```
SQL>
/*****
SQL> /* Step 10 - SHUTDOWN THE DATABASE..!!!!
SQL> */
SQL>
/*****
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
SQL>
SQL>
SQL> DOC
DOC>#####
DOC>#####
DOC>
DOC> The above sql script is the final step of the upgrade. Please
DOC> review any errors in the spool log file. If there are any errors
in
DOC> the spool file, consult the Oracle Database Upgrade Guide for
```

```
DOC> troubleshooting recommendations.
DOC>
DOC> Next restart for normal operation, and then run utlrlp.sql to
DOC> recompile any invalid application objects.
DOC>
DOC>#####
DOC>#####
DOC>#
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> Rem Set errorlogging off
SQL> SET ERRORLOGGING OFF;
SP2-0735: unknown SET option beginning "ERRORLOGGI..."
SQL>
SQL> Rem
*****
SQL> Rem END catupgrd.sql
SQL> Rem
*****
```

Después de la ejecución de `catupgrd.sql` volvemos a tener la base de datos caída. Es por esto que debemos volver a levantarla para reinicializar los parámetros del sistema para que vuelvan a operar de una manera normal. Esta **reinicialización** posterior a la ejecución de `catupgrd.sql` nivela las cachés, limpia los buffers y realiza otras tareas de limpieza.

Estas medidas son un paso final importante para asegurar la integridad y la consistencia del nuevo software de la base de datos actualizado. Así pues ejecutamos lo siguiente:

```
SQL> sqlplus /nolog
SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area  347348992 bytes
Fixed Size                  1299820 bytes
Variable Size              130026132 bytes
Database Buffers          209715200 bytes
Redo Buffers                6307840 bytes
Database mounted.
Database opened.
```

Una vez levantada de nuevo la base de datos ejecutamos el script de **Post-Upgrade Status Tool**, `utlu111s.sql`, que proporciona un resumen de todo lo ejecutado durante la migración. Muestra el estado de los componentes de la base de datos actualizada y el tiempo requerido para completar el upgrade de cada uno de ellos. Cualquier error que se haya producido durante el upgrade es listado con cada componente. Este script se encuentra en `$ORACLE_HOME/rdbms/admin`, por lo que situados allí ejecutamos:

```
SQL> @utlu111s.sql
```

```
Oracle Database 11.1 Post-Upgrade Status Tool          10-15-2009
18:29:59
.
Component                               Status          Version  HH:MM:SS
.
Oracle Server                            VALID          11.1.0.6.0  00:41:29
JServer JAVA Virtual Machine             VALID          11.1.0.6.0  00:07:35
Oracle Workspace Manager                 VALID          10.2.0.4.3  00:00:02
OLAP Analytic Workspace                  VALID          11.1.0.6.0  00:01:00
OLAP Catalog                             VALID          11.1.0.6.0  00:02:08
Oracle OLAP API                          VALID          11.1.0.6.0  00:00:52
Oracle Enterprise Manager                VALID          11.1.0.6.0  00:25:21
Oracle XDK                               VALID          11.1.0.6.0  00:01:38
Oracle Text                              VALID          11.1.0.6.0  00:02:03
Oracle XML Database                      VALID          11.1.0.6.0  00:08:08
Oracle Database Java Packages            VALID          11.1.0.6.0  00:00:46
Oracle Multimedia                        VALID          11.1.0.6.0  00:08:44
Spatial                                  VALID          11.1.0.6.0  00:13:32
Oracle Expression Filter                 VALID          11.1.0.6.0  00:00:31
Oracle Rule Manager                      VALID          11.1.0.6.0  00:00:28
Gathering Statistics                     00:10:33

Total Upgrade Time: 02:05:05

PL/SQL procedure successfully completed.
```

Como se puede comprobar en la salida de la ejecución del script, no hay **ningún mensaje de error ni objeto inválido**, por lo que podemos seguir adelante sin problemas. Además, a modo informativo podemos ver también los tiempos de upgrade que han sido necesarios para cada componente y que **el total ha sido de 2 horas y 5 minutos**.

El siguiente paso es **ejecutar el script catuppst.sql**, situado en el mismo directorio donde nos encontramos. Este script realiza acciones de actualización (upgrade) de componentes y objetos que no necesitan estar en UPGRADE MODE:

```
SQL> @catuppst.sql
```

```
TIMESTAMP
-----
COMP_TIMESTAMP POSTUP_BGN 2009-10-15 18:36:00
PL/SQL procedure successfully completed.
This script will migrate the Baseline data on a pre-11g database to
the 11g database.
Move BL Data SYS.WRH$_FILESTATXS
Move BL Data SYS.WRH$_SQLSTAT
Move BL Data SYS.WRH$_SYSTEM_EVENT
Move BL Data SYS.WRH$_WAITSTAT
Move BL Data SYS.WRH$_LATCH
Move BL Data SYS.WRH$_LATCH_CHILDREN
Move BL Data SYS.WRH$_LATCH_PARENT
Move BL Data SYS.WRH$_LATCH_MISSES_SUMMARY
Move BL Data SYS.WRH$_DB_CACHE_ADVICE
Move BL Data SYS.WRH$_ROWCACHE_SUMMARY
```

```
Move BL Data SYS.WRH$_SGASTAT
Move BL Data SYS.WRH$_SYSSTAT
Move BL Data SYS.WRH$_PARAMETER
Move BL Data SYS.WRH$_SEG_STAT
Move BL Data SYS.WRH$_DLM_MISC
Move BL Data SYS.WRH$_SERVICE_STAT
Move BL Data SYS.WRH$_TABLESPACE_STAT
Move BL Data SYS.WRH$_OSSTAT
Move BL Data SYS.WRH$_SYS_TIME_MODEL
Move BL Data SYS.WRH$_SERVICE_WAIT_CLASS
Move BL Data SYS.WRH$_INST_CACHE_TRANSFER
Move BL Data SYS.WRH$_ACTIVE_SESSION_HISTORY
...
... Completed Moving the Baseline Data ...
...
... If there are no Move BL Data messages ...
... above, then there are no renamed ...
... baseline tables in the system. ...
...
Drop Renamed Baseline Table SYS.WRH$_FILESTATXS_BR
Drop Renamed Baseline Table SYS.WRH$_SQLSTAT_BR
Drop Renamed Baseline Table SYS.WRH$_SYSTEM_EVENT_BR
Drop Renamed Baseline Table SYS.WRH$_WAITSTAT_BR
Drop Renamed Baseline Table SYS.WRH$_LATCH_BR
Drop Renamed Baseline Table SYS.WRH$_LATCH_CHILDREN_BR
Drop Renamed Baseline Table SYS.WRH$_LATCH_PARENT_BR
Drop Renamed Baseline Table SYS.WRH$_LATCH_MISSES_SUMMARY_BR
Drop Renamed Baseline Table SYS.WRH$_DB_CACHE_ADVICE_BR
Drop Renamed Baseline Table SYS.WRH$_ROWCACHE_SUMMARY_BR
Drop Renamed Baseline Table SYS.WRH$_SGASTAT_BR
Drop Renamed Baseline Table SYS.WRH$_SYSSTAT_BR
Drop Renamed Baseline Table SYS.WRH$_PARAMETER_BR
Drop Renamed Baseline Table SYS.WRH$_SEG_STAT_BR
Drop Renamed Baseline Table SYS.WRH$_DLM_MISC_BR
Drop Renamed Baseline Table SYS.WRH$_SERVICE_STAT_BR
Drop Renamed Baseline Table SYS.WRH$_TABLESPACE_STAT_BR
Drop Renamed Baseline Table SYS.WRH$_OSSTAT_BR
Drop Renamed Baseline Table SYS.WRH$_SYS_TIME_MODEL_BR
Drop Renamed Baseline Table SYS.WRH$_SERVICE_WAIT_CLASS_BR
Drop Renamed Baseline Table SYS.WRH$_INST_CACHE_TRANSFER_BR
Drop Renamed Baseline Table SYS.WRH$_ACTIVE_SESSION_HISTORY_BR
...
... Completed the Dropping of the ...
... Renamed Baseline Tables ...
...
... If there are no Drop Table messages ...
... above, then there are no renamed ...
... baseline tables in the system. ...
...
PL/SQL procedure successfully completed.
39 rows created.
Commit complete.
Table created.
```

```
3 rows created.
1 row updated.
3 rows updated.
39 rows updated.
Table dropped.
Commit complete.
39 rows updated.
Commit complete.
39 rows updated.
Commit complete.
21 rows updated.
Commit complete.
21 rows created.
Commit complete.
9 rows created.
Commit complete.
PL/SQL procedure successfully completed.
TIMESTAMP
-----
COMP_TIMESTAMP POSTUP_END 2009-10-15 18:36:15
```

Ahora **ejecutamos el script utlrp.sql**, que se encuentra en el mismo lugar que los anteriores. Este script se encarga de recompilar cualquier PL/SQL almacenado y cualquier código java que estuviera en otra sesión:

```
SQL> @utlrp.sql
```

```
TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_BGN 2009-10-15 18:38:15
DOC> The following PL/SQL block invokes UTL_RECOMP to recompile
invalid
DOC> objects in the database. Recompilation time is proportional to
the
DOC> number of invalid objects in the database, so this command may
take
DOC> a long time to execute on a database with a large number of
invalid
DOC> objects.
DOC>
DOC> Use the following queries to track recompilation progress:
DOC>
DOC> 1. Query returning the number of invalid objects remaining.
This
DOC> number should decrease with time.
DOC> SELECT COUNT(*) FROM obj$ WHERE status IN (4, 5, 6);
```

```
DOC>
DOC> 2. Query returning the number of objects compiled so far. This
number
DOC> should increase with time.
DOC> SELECT COUNT(*) FROM UTL_RECOMP_COMPILED;
DOC>
DOC> This script automatically chooses serial or parallel
recompilation
DOC> based on the number of CPUs available (parameter cpu_count)
multiplied
DOC> by the number of threads per CPU (parameter
parallel_threads_per_cpu).
DOC> On RAC, this number is added across all RAC nodes.
DOC>
DOC> UTL_RECOMP uses DBMS_SCHEDULER to create jobs for parallel
recompilation. Jobs are created without instance affinity so
that they
DOC> can migrate across RAC nodes. Use the following queries to
verify
DOC> whether UTL_RECOMP jobs are being created and run correctly:
DOC>
DOC> 1. Query showing jobs created by UTL_RECOMP
DOC> SELECT job_name FROM dba_scheduler_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC> SELECT job_name FROM dba_scheduler_running_jobs
DOC> WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#
```

PL/SQL procedure successfully completed.

TIMESTAMP

COMP_TIMESTAMP UTLRP_END 2009-10-15 19:01:47

PL/SQL procedure successfully completed.

```
DOC> The following query reports the number of objects that have
compiled
DOC> with errors (objects that compile with errors have status set to
3 in
DOC> obj$). If the number is higher than expected, please examine the
error
DOC> messages reported with each object (using SHOW ERRORS) to see if
they
DOC> point to system misconfiguration or resource constraints that
must be
DOC> fixed before attempting to recompile these objects.
DOC>#
```

OBJECTS WITH ERRORS

0

```
DOC> The following query reports the number of errors caught during
```

```
DOC> recompilation. If this number is non-zero, please query the
error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these
errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#
```

```
ERRORS DURING RECOMPILATION
```

```
-----
                                0
```

```
PL/SQL procedure successfully completed.
```

```
PL/SQL procedure successfully completed.
```

Por último **ejecutaremos el script dbupgdiag.sql** para comprobar la integridad de la base de datos. Este script ya lo utilizamos antes de iniciar la migración con el mismo objetivo. Recordar que el script lo teníamos localizado en \$HOME/migra y que el resultado de la ejecución se volcaba en un archivo, en este caso db_upg_diag_DBase10_15-Oct-2009_0739.log. Ejecutamos:

```
SQL> @ $HOME/migra dbupgdiag.sql
```

El contenido de db_upg_diag_DBase10_15-Oct-2009_0739.log es el siguiente:

```
*** Start of LogFile ***

Oracle Database Upgrade Diagnostic Utility      10-15-2009 19:39:10

=====
Database Uptime
=====

19:38 15-OCT-09

=====
Database Wordsize
=====

This is a 32-bit database

=====
Software Verison
=====

Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
PL/SQL Release 11.1.0.6.0 - Production
CORE 11.1.0.6.0 Production
TNS for Linux: Version 11.1.0.6.0 - Production
NLSRTL Version 11.1.0.6.0 - Production

=====
Compatibility
=====
```


Compatibility is set as 10.2.0.4

=====
Component Status
=====

Comp ID	Component	Status	Version	Org_Version
AMD	OLAP Catalog	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
APS	OLAP Analytic Workspace	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
CATALOG	Oracle Database Catalog Views	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.1.0				
CATJAVA	Oracle Database Java Packages	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
CATPROC	Oracle Database Packages and Types	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
CONTEXT	Oracle Text	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
EM	Oracle Enterprise Manager	VALID	11.1.0.6.0	10.2.0.4.0
EXF	Oracle Expression Filter	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
JAVAVM	JServer JAVA Virtual Machine	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
ODM	Oracle Data Mining	VALID	11.1.0.6.0	10.2.0.4.0
ORDIM	Oracle Multimedia	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
OWM	Oracle Workspace Manager	VALID	10.2.0.4.3	10.2.0.1.0
RUL	Oracle Rule Manager	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
SDO	Spatial	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
XDB	Oracle XML Database	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
XML	Oracle XDK	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				
XOQ	Oracle OLAP API	VALID	11.1.0.6.0	10.2.0.1.0
10.2.0.4.0				

=====
List of Invalid Database Objects Owned by SYS / SYSTEM
=====

Number of Invalid Objects

There are no Invalid Objects

```
DOC>#####  
DOC>  
DOC> If there are no Invalid objects below will result in zero rows.  
DOC>  
DOC>#####  
DOC>#
```

no rows selected

=====
List of Invalid Database Objects
=====

Number of Invalid Objects

There are no Invalid Objects

```
DOC>#####  
DOC>  
DOC> If there are no Invalid objects below will result in zero rows.  
DOC>  
DOC>#####  
DOC>#
```

no rows selected

=====
Identifying whether a database was created as 32-bit or 64-bit
=====

```
DOC>#####  
DOC>  
DOC> Result referencing the string 'B023' ==> Database was created as 32-bit  
DOC> Result referencing the string 'B047' ==> Database was created as 64-bit  
DOC> When String results in 'B023' and when upgrading database to 10.2.0.3.0  
DOC> (64-bit) , For known issue refer below articles  
DOC>  
DOC> Note 412271.1 ORA-600 [22635] and ORA-600 [KOKEIIX1] Reported While  
DOC> Upgrading Or Patching Databases To 10.2.0.3  
DOC> Note 579523.1 ORA-600 [22635], ORA-600 [KOKEIIX1], ORA-7445 [KOPESIZ] and  
DOC> OCI-21500 [KOXSIHREAD1] Reported While Upgrading To 11.1.0.6  
DOC>  
DOC>#####  
DOC>#
```

Metadata Initial DB Creation Info

B023 Database was created as 32-bit

=====
Number of Duplicate Objects Owned by SYS and SYSTEM
=====

Counting duplicate objects

```
COUNT(1)  
-----  
4
```

=====
Duplicate Objects Owned by SYS and SYSTEM
=====

Querying duplicate objects

OBJECT_NAME	OBJECT_TYPE
-----	-----
AQ\$_SCHEDULES	TABLE
AQ\$_SCHEDULES_PRIMARY	INDEX
DBMS_REPCAT_AUTH	PACKAGE
DBMS_REPCAT_AUTH	PACKAGE BODY

```
DOC>  
DOC>#####  
DOC>  
DOC> If any objects found please follow below article.  
DOC> Note 1030426.6 How to Clean Up Duplicate Objects Owned by SYS and SYSTEM schema
```

```
DOC> Read the Exceptions carefully before taking actions.
DOC>
DOC>#####
DOC>#

=====
JVM Verification
=====

=====
Checking Existence of Java-Based Users and Roles
=====

DOC>
DOC>#####
DOC>
DOC> There should not be any Java Based users for database version 9.0.1 and above.
DOC> If any users found, it is faulty JVM.
DOC>
DOC>#####
DOC>#

User Existence
-----
No Java Based Users

DOC>
DOC>#####
DOC>
DOC> Healthy JVM Should contain Six Roles.
DOC> If there are more or less than six role, JVM is inconsistent.
DOC>
DOC>#####
DOC>#
Role
-----
There are 6 JAVA related roles

Roles

ROLE
-----
JAVAUSERPRIV
JAVAIDPRIV
JAVASYSPRIV
JVADEBUGPRIV
JAVA_ADMIN
JAVA_DEPLOY

=====
List of Invalid Java Objects owned by SYS
=====

There are no SYS owned invalid JAVA objects

DOC>
DOC>#####
DOC>
DOC> Check the status of the main JVM interface packages DBMS_JAVA
DOC> and INITJVMAUX and make sure it is VALID.
DOC> If there are no Invalid objects below will result in zero rows.
DOC>
DOC>#####
DOC>#
```

```
no rows selected
INFO: Below query should succeed with 'foo' as result.

JAVAVM TESTING
-----
foo

*** End of LogFile ***
```

Y hasta aquí llega el paso 5, hacer el upgrade de la base de datos. El **upgrade/migración ha sido completado con éxito** y ahora debemos realizar los ajustes necesarios para finalizar todo el proceso.

Paso 6 - Ajuste de la base de datos

En este último paso vamos a realizar una serie de tareas con el fin de ajustar al máximo el rendimiento de nuestra nueva base de datos. Después de una migración es necesario realizar algunos procesos para completar correctamente el upgrade ya que debemos ajustar el antiguo sistema a la nueva forma de trabajar instalada. Además debemos hacer otras tareas como reinicializar componentes que hemos parado.

Una vez hecha la migración cabe recordar que, a partir de este momento, la variable **ORACLE_HOME** ya no es /u01/app/oracle/product/10.2.0.4/ ahora debe ser /u01/app/oracle/product/11.1.0/db_1. Igualmente la variable **PATH** debe haber cambiado para albergar los archivos y ejecutables referentes a la nueva base de datos. Esto ya lo hemos comprobado en pasos anteriores, sin embargo mostramos su estado actual:

```
[oracle@nod01 ~]$ echo $ORACLE_HOME
/u01/app/oracle/product/11.1.0/db_1
[oracle@nod01 ~]$ echo $ORACLE_BASE
/u01/app/oracle
[oracle@nod01 ~]$ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/usr/bin:/bin:/usr/X
11R6/bin:/usr/java/jre1.5.0_10/bin:/home/oracle/bin:/
sbin:/usr/sbin:/u01/app/oracle/product/11.1.0/db_1/
bin:/usr/java/jre1.5.0_10/bin:/usr/java/jre1.5.0_10/
bin
```

Del mismo modo comprobamos que el archivo **oratab** contiene la nueva referencia del **ORACLE_HOME** correctamente y la **actualizamos para volver a habilitar el arranque automático** cambiando la N final por una Y:

```
DBase10:/u01/app/oracle/product/11.1.0/db_1:Y
```

Comprobamos que el nuevo archivo **listener.ora**, que fue generado en el paso anterior, se encuentra en el directorio que le corresponde y está configurado correctamente:

```
[oracle@nod01 admin]$ pwd
/u01/app/oracle/product/11.1.0/db_1/network/admin/
[oracle@nod01 admin]$ ls
listener.ora  samples  shrept.lst  tnsnames.ora
```

Ahora levantamos el listener mediante:

```
[oracle@nod01 db_1]$ lsnrctl start
LSNRCTL for Linux: Version 11.1.0.6.0 - Production on
16-OCT-2009 13:40:19
Copyright (c) 1991, 2007, Oracle. All rights
reserved.
Starting
/u01/app/oracle/product/11.1.0/db_1/bin/tnslsnr:
please wait...
TNSLSNR for Linux: Version 11.1.0.6.0 - Production
System parameter file is
/u01/app/oracle/product/11.1.0/db_1/network/admin/li
stener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/myacshost/listener/aler
t/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=myacshost.
us.oracle.com)(PORT=1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521
)))
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=myacshost.
us.oracle.com)(PORT=1521)))
STATUS of the LISTENER
-----
Alias                          LISTENER
Version                        TNSLSNR for Linux: Version
11.1.0.6.0 - Production
Start Date                     16-OCT-2009 13:40:21
Uptime                         0 days 0 hr. 0 min. 0 sec
Trace Level                    off
Security                       ON: Local OS Authentication
SNMP                           OFF
Listener Parameter File
/u01/app/oracle/product/11.1.0/db_1/network/admin/li
stener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/myacshost/listener/aler
t/log.xml
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=myacshost.u
s.oracle.com)(PORT=1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)
))
The listener supports no services
The command completed successfully
```

Ahora comprobamos que la base de datos está trabajando sin spfile, tal y como es esperado, ya que creamos un pfile con los cambios necesarios para la nueva versión en un paso anterior de la migración. Comprobamos si existe mediante la siguiente ejecución:

```
SQL> show parameter spfile
NAME                                TYPE                                VALUE
-----                                -                                -
spfile                               string
```

Tras comprobar que, efectivamente, no existe el spfile **creamos un archivo de arranque spfile** a partir del pfile que creamos en un paso anterior y que está situado en \$ORACLE_HOME/dbs. Esto lo hacemos, conectados a la base de datos, mediante:

```
SQL> create spfile from pfile;
File created.
```

Esta ejecución crea el archivo **spfileDBase10.ora** y será el que utilice la base de datos a partir de ahora.

El último paso que vamos a realizar es el **bloqueo de todas las cuentas de usuario suministradas excepto SYS y SYSTEM y expirar sus contraseñas**. Esto es una recomendación de Oracle ya que, dependiendo desde qué versión de la base de datos se realiza el upgrade, puede haber nuevas cuentas suministradas.

Antes de realizar el bloqueo consultamos el estado de las cuentas mediante la ejecución de la siguiente sentencia SQL:

```
SQL> SELECT username, account_status FROM dba_users
       ORDER BY username;
USERNAME                                ACCOUNT_STATUS
-----                                -
ANONYMOUS                               EXPIRED & LOCKED
APEX_PUBLIC_USER                         EXPIRED & LOCKED
BI                                       EXPIRED & LOCKED
CTXSYS                                   EXPIRED & LOCKED
DBSNMP                                   OPEN
DIP                                       EXPIRED & LOCKED
EXFSYS                                   EXPIRED & LOCKED
FLOWS_030000                             EXPIRED & LOCKED
FLOWS_FILES                              EXPIRED & LOCKED
HR                                       EXPIRED & LOCKED
IX                                       EXPIRED & LOCKED

USERNAME                                ACCOUNT_STATUS
-----                                -
MDDATA                                   EXPIRED & LOCKED
MDSYS                                    EXPIRED & LOCKED
MGMT_VIEW                                OPEN
MIGUEL                                    OPEN
OE                                       EXPIRED & LOCKED
OLAPSYS                                   EXPIRED & LOCKED
ORACLE_OCM                               EXPIRED & LOCKED
ORDPLUGINS                               EXPIRED & LOCKED
ORDSYS                                   EXPIRED & LOCKED
OUTLN                                    EXPIRED & LOCKED
OWBSYS                                   EXPIRED & LOCKED

USERNAME                                ACCOUNT_STATUS
-----                                -
PM                                       EXPIRED & LOCKED
SCOTT                                    OPEN
SH                                       EXPIRED & LOCKED
SI_INFORMTN_SCHEMA                       EXPIRED & LOCKED
```

```
SPATIAL_CSW_ADMIN_USR      EXPIRED & LOCKED
SPATIAL_WFS_ADMIN_USR     EXPIRED & LOCKED
SYS                         OPEN
SYSMAN                     OPEN
SYSTEM                     OPEN
TSM SYS                    EXPIRED & LOCKED
WKPROXY                    EXPIRED & LOCKED
```

```
-----
USERNAME                    ACCOUNT_STATUS
-----
WKSYS                      EXPIRED & LOCKED
WK_TEST                    EXPIRED & LOCKED
WMSYS                      EXPIRED & LOCKED
XDB                        EXPIRED & LOCKED
XS$NULL                    EXPIRED & LOCKED
```

38 rows selected.

Para bloquear las cuentas de usuario y expirar sus contraseñas ejecutamos:

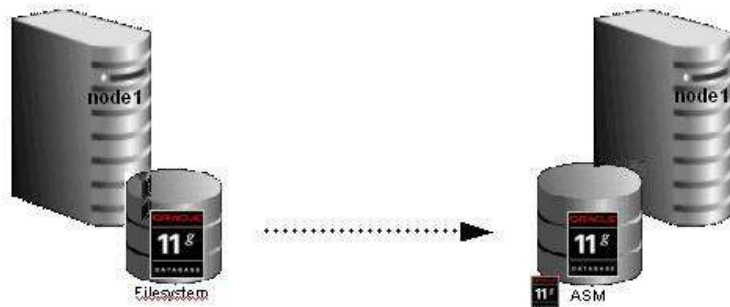
```
SQL> ALTER USER DBSNMP PASSWORD EXPIRE ACCOUNT LOCK;
User altered.
```

Esta sentencia la ejecutamos una vez para cada usuario que deseamos bloquear.

Y tras esta última ejecución **damos por finalizado el proceso de migración/upgrade de la base de datos**. Actualmente nuestra base de datos es una 11.1.0.6 y trabaja con normalidad.

Cambiar la single instance 11.1.0.6 a ASM con RMAN

Una vez hecha la migración de nuestra base de datos es el momento de **realizar el cambio de tipo de almacenamiento**. Actualmente tenemos nuestra base de datos trabajando en File System y vamos a realizar el paso a Automatic Storage Management (ASM).



Antes de empezar cabe recordar que **la instalación de ASM que hemos realizado en un paso anterior la hemos hecho en cluster**, es decir, cuando cambiemos el almacenamiento de file system a ASM ya tendremos el almacenamiento preparado para RAC. No habrá que hacer nuevas configuraciones a este respecto.

Para realizar esta tarea lo primero que debemos hacer es **editar el init.ora** para que apunte al nuevo control file que va a tener la base de datos:

```
control_files =  
'+DATA/control01.ctl','+FRA/control02.ctl','+FRA/con  
trol03.ctl'
```

Ahora, debemos **rearrancar la base de datos en modo nomount**. Esto lo hacemos mediante:

```
SQL> shutdown immediate  
Database closed.  
Database dismounted.  
ORACLE instance shut down.  
SQL> startup nomount
```

Tras esto, debemos conectarnos mediante una sesión de RMAN para realizar una **copia de los control files** (son 3 archivos) **y llevarlos a su nueva localización**. Así pues, nos conectamos a RMAN:

```
[oracle@nod01 ~]$ rman  
Recovery Manager: Release 11.1.0.6.0 - Production on  
Tue Nov 3 17:35:37 2009  
Copyright (c) 1982, 2007, Oracle. All rights  
reserved.
```

```
RMAN> connect target  
connected to target database: DBASE10 (not mounted)
```

Realizamos la copia de los control files ejecutando:

```
RMAN> RESTORE CONTROLFILE FROM  
'/u01/app/oracle/oradata/DBase10/control02.ctl';  
Starting restore at 03-NOV-09  
using target database control file instead of  
recovery catalog  
allocated channel: ORA_DISK_1  
channel ORA_DISK_1: SID=147 device type=DISK  
channel ORA_DISK_1: copied control file copy  
output file  
name=/u01/app/oracle/oradata/DBase10/control01.ctl  
output file  
name=/u01/app/oracle/oradata/DBase10/control02.ctl  
output file  
name=/u01/app/oracle/oradata/DBase10/control03.ctl  
Finished restore at 03-NOV-09
```

```
RMAN> RESTORE CONTROLFILE FROM  
'/u01/app/oracle/oradata/DBase10/control01.ctl';  
Starting restore at 03-NOV-09  
using channel ORA_DISK_1  
channel ORA_DISK_1: copied control file copy  
output file  
name=/u01/app/oracle/oradata/DBase10/control01.ctl  
output file  
name=/u01/app/oracle/oradata/DBase10/control02.ctl  
output file  
name=/u01/app/oracle/oradata/DBase10/control03.ctl  
Finished restore at 03-NOV-09
```



```
RMAN> RESTORE CONTROLFILE FROM
'/u01/app/oracle/oradata/DBase10/control03.ctl';
Starting restore at 03-NOV-09
using channel ORA_DISK_1
channel ORA_DISK_1: copied control file copy
output file
  name=/u01/app/oracle/oradata/DBase10/control01.ctl
output file
  name=/u01/app/oracle/oradata/DBase10/control02.ctl
output file
  name=/u01/app/oracle/oradata/DBase10/control03.ctl
Finished restore at 03-NOV-09
```

Antes de continuar debemos poner la base de datos de nuevo en **modo mount**:

```
SQL> alter database mount;
```

El siguiente paso es realizar una **copia de los datafiles y situarlos en ASM**. Esto lo realizamos mediante la siguiente ejecución en RMAN:

```
RMAN> BACKUP AS COPY DATABASE FORMAT '+DATA';
Starting backup at 03-NOV-09
released channel: ORA_DISK_1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=147 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001
  name=/u01/app/oracle/oradata/DBase10/system01.dbf
output file
  name=+DATA/dbase10/datafile/system.256.701977649
  tag=TAG20091103T174721 RECID=1 STAMP=701977758
channel ORA_DISK_1: datafile copy complete, elapsed
  time: 00:02:10
channel ORA_DISK_1: starting datafile copy
input datafile file number=00002
  name=/u01/app/oracle/oradata/DBase10/sysaux01.dbf
output file
  name=+DATA/dbase10/datafile/sysaux.257.701977775
  tag=TAG20091103T174721 RECID=2 STAMP=701977847
channel ORA_DISK_1: datafile copy complete, elapsed
  time: 00:01:15
channel ORA_DISK_1: starting datafile copy
input datafile file number=00003
  name=/u01/app/oracle/oradata/DBase10/undotbs01.dbf
output file
  name=+DATA/dbase10/datafile/undotbs1.258.701977851
  tag=TAG20091103T174721 RECID=3 STAMP=701977853
channel ORA_DISK_1: datafile copy complete, elapsed
  time: 00:00:07
channel ORA_DISK_1: starting datafile copy
copying current control file
output file
  name=+DATA/dbase10/controlfile/backup.259.701977857
  tag=TAG20091103T174721 RECID=4 STAMP=701977859
```

```
channel ORA_DISK_1: datafile copy complete, elapsed
time: 00:00:03
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004
name=/u01/app/oracle/oradata/DBase10/users01.dbf
output file
name=+DATA/dbase10/datafile/users.260.701977861
tag=TAG20091103T174721 RECID=5 STAMP=701977861
channel ORA_DISK_1: datafile copy complete, elapsed
time: 00:00:01
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup
set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 03-NOV-09
channel ORA_DISK_1: finished piece 1 at 03-NOV-09
piece
handle=+DATA/dbase10/backupset/2009_11_03/nnsnf0_tag
20091103t174721_0.261.701977863
tag=TAG20091103T174721 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed
time: 00:00:01
Finished backup at 03-NOV-09
```

Ahora renombramos los datafiles mediante RMAN ejecutando:

```
RMAN> SWITCH DATABASE TO COPY;
datafile 1 switched to datafile copy
"+DATA/dbase10/datafile/system.256.701977649"
datafile 2 switched to datafile copy
"+DATA/dbase10/datafile/sysaux.257.701977775"
datafile 3 switched to datafile copy
"+DATA/dbase10/datafile/undotbs1.258.701977851"
datafile 4 switched to datafile copy
"+DATA/dbase10/datafile/users.260.701977861"
```

Una vez hecho esto debemos realizar un **switch del tempfile** mediante:

```
RMAN> run{
2> set newname for tempfile 1 to '+DATA';
3> switch tempfile all;
4> }
executing command: SET NEWNAME
renamed tempfile 1 to +DATA in control file
```

Tras esta última ejecución **abrimos la base de datos** y salimos de RMAN:

```
RMAN> ALTER DATABASE OPEN;
database opened
```

Tras estas operaciones comprobamos cuáles son ahora los datafiles de la base de datos:

```
SQL> select FILE#, STATUS, NAME from v$datafile;
FILE# STATUS
-----
NAME
```

```
-----  
1 SYSTEM  
+DATA/dbase11/datafile/system.256.701977649  
  
2 ONLINE  
+DATA/dbase11/datafile/sysaux.257.701977775  
  
3 ONLINE  
+DATA/dbase11/datafile/undotbs1.258.701977851  
  
4 ONLINE  
+DATA/dbase11/datafile/users.260.701977861
```

El último paso que nos queda por dar para finalizar la migración de File System a ASM es el **cambio de los archivos de redo**. Para ello deberemos añadir nuevos redo logs a la base de datos en ASM para que reemplacen a los actuales, ejecutar un cambio de redo logs y eliminar los antiguos.

Consultamos los archivos de redo log actuales:

```
SQL> SELECT a.group#, b.member, a.status FROM v$log  
a, v$logfile b WHERE a.group#=b.group#;  
GROUP#  MEMBER  
-----  
STATUS  
-----  
3          /u01/app/oracle/oradata/DBase10/redo03.log  
INACTIVE  
2          /u01/app/oracle/oradata/DBase10/redo02.log  
INACTIVE  
1          /u01/app/oracle/oradata/DBase10/redo01.log  
CURRENT
```

Como se puede observar, el redo log que está actualmente en uso es el redo01.log, como indica la consulta mediante su STATUS CURRENT.

Como se puede observar en la consulta mostrada, nuestra base de datos actual tiene 3 grupos de redo con un redo log cada uno.

Ahora **añadimos un redo logfile a cada grupo** de redo mediante:

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER '+DATA'TO  
GROUP 1;  
Database altered.
```

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER '+DATA'TO  
GROUP 2;  
Database altered.
```

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER '+DATA'TO  
GROUP 3;  
Database altered.
```

```
SQL> ALTER DATABASE DROP LOGFILE MEMBER  
'/u01/app/oracle/oradata/DBase10/redo02.log';
```

Database altered.

Después de añadir los archivos debemos realizar varias operaciones. Primero, teniendo en cuenta que el redo log que está actualmente en uso es el redo01.log, realizamos un switch consiguiendo **cambiar al nuevo redo log que hemos creado en +DATA, en el grupo 2:**

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
System altered.
```

Esto lo hacemos así ya que redo02.log ya ha sido eliminado y con el switch cambiamos al archivo que hemos creado antes. De esta forma ya estamos trabajando con los nuevos logs en ASM.

Tras este cambio podemos eliminar el redo03.log y el redo01.log ya que no están en uso dejando así **en los grupos 1 y 3 un solo archivo de log.**

```
SQL> ALTER DATABASE DROP LOGFILE MEMBER
'/u01/app/oracle/oradata/DBase10/redo03.log';
Database altered.
```

```
SQL> ALTER DATABASE DROP LOGFILE MEMBER
'/u01/app/oracle/oradata/DBase10/redo01.log';
Database altered
```

Ahora, una vez hechas estas operaciones cambiamos de logfile por última vez para comprobar que funcionan todos correctamente.

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
System altered.
```

Hacemos una consulta para ver el **estado final de los grupos de log:**

```
SQL> select * from v$log;
  GROUP#    THREAD#  SEQUENCE#    BYTES
-----  -
MEMBERS ARC STATUS
-----  -
FIRST_CHANGE# FIRST_TIM
-----  -
      1          1          7  52428800
1 NO ACTIVE
  614424 06-NOV-09
      2          1          8  52428800
2 NO ACTIVE
  614594 06-NOV-09
      3          1          9  52428800
2 NO CURRENT
  614698 06-NOV-09
```

Consultamos el nombre y la localización de los nuevos archivos de log y **comprobamos que se han eliminado los antiguos:**

```
SQL> select * from v$logfile;
```

```
  GROUP# STATUS TYPE
-----  -
MEMBER
-----  -
```

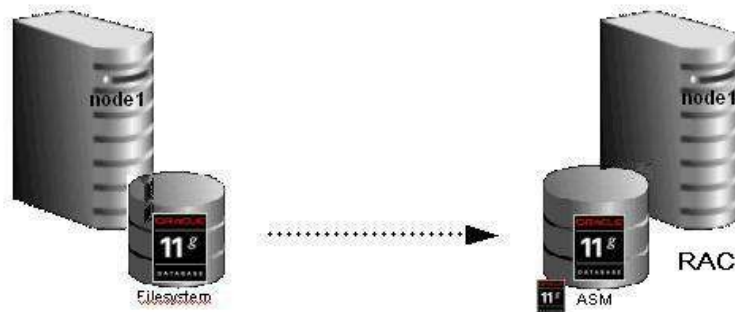
```
IS_
---
          1          ONLINE
+DATA/dbase10/onlinelog/group_1.263.701979411
NO
          2          ONLINE
+DATA/dbase10/onlinelog/group_2.265.701981023
NO
          3          ONLINE
+DATA/dbase10/onlinelog/group_3.267.702213275
NO
```

Con esta última consulta comprobamos que está todo correcto y damos por **finalizada esta configuración**.

En este momento tenemos un sistema con una base de datos 11.1.0.6 con ASM funcionando sin problemas.

Convertir la single instance 11.1.0.6 en una base de datos RAC 11.1.0.6

En este punto acometemos el último paso en la migración de nuestro sistema. En este momento disponemos de un sistema single-instance en ASM y Clusterware instalado. Ahora, **crearemos nuestro RAC de un nodo**.



Esta ejecución la dividimos en varias partes:

Creación de un listener para RAC y configuración del sistema

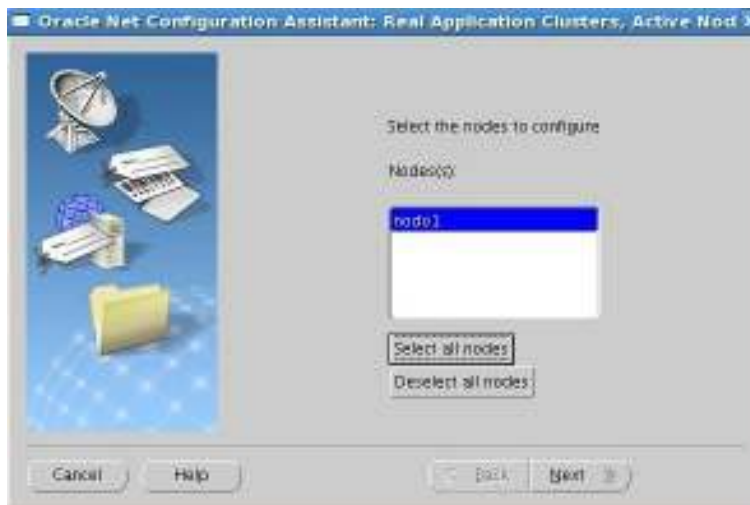
El primer paso para configurar nuestro RAC es **crear un listener** para él. Esto lo hacemos mediante Network Configuration Assistant de Oracle, netca:

```
[oracle@nod01 ~]$ netca
```

En la pantalla de inicio de Network Configuration Assistant elegimos la opción *Cluster Configuration*:



Seleccionamos nuestro **nodo1** para configurar el listener en él:



En la siguiente pantalla seleccionamos *Listener Configuration*:



Seleccionamos *Add* para añadir uno nuevo:



Oracle recomienda poner como nombre del listener LISTENER. NETCA añadirá automáticamente `_NODO1` al nombre de éste cuando lo cree. Si no fuera así no podríamos darle este nombre ya que ya tenemos configurado un listener con nombre LISTENER que da servicio a la base de datos y a ASM:

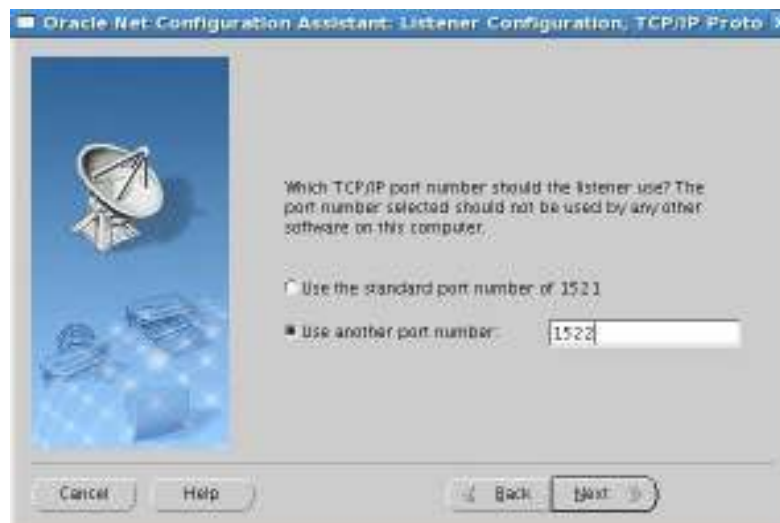


Así pues, luego lo comprobaremos, el listener que configuramos es llamado LISTENER_NODO1.

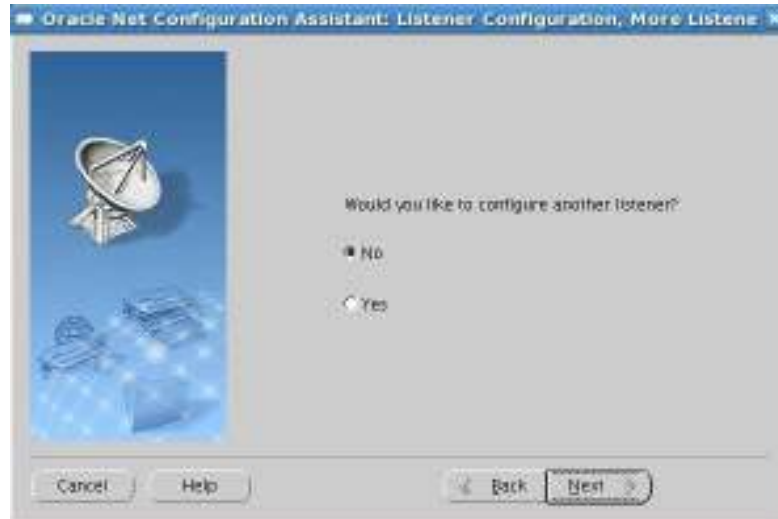
Seleccionamos el **protocolo TCP** solamente y clicamos sobre *Next*:



Al igual que pasaba antes con el nombre del listener, se recomienda utilizar el puerto 1521, sin embargo ya está ocupado por el listener anterior, por lo tanto elegimos el 1522:



Nos aparece ahora la pregunta de si queremos configurar otro listener, respondemos *No* y continuamos:



Ahora seleccionamos el listener que hemos creado y lo **arrancamos**:



Por último, nos aparece una pantalla en la que se nos muestra un mensaje confirmando que se ha realizado la **configuración con éxito**:



Tras la configuración del listener **comprobamos el estado de los servicios de nuestro Clusterware:**

```
[oracle@nod01 admin]$ srvctl status nodeapps -n nod01
VIP is running on node: nod01
GSD is running on node: nod01
Listener is running on node: nod01
ONS daemon is running on node: nod01
```

Ahora mostramos como ha quedado configurado el archivo **listener.ora:**

```
LISTENER_NOD01 =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = nod01-
vip) (PORT = 1521))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST =
192.168.160.128) (PORT = 1521))
      )
    )
  )

SID_LIST_LISTENER_NOD01 =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME =
/u01/app/oracle/product/11.1.0/db_1)
      (PROGRAM = extproc)
    )
  )
```

Igualmente, mostramos el estado final de **tnsnames.ora:**

```
LISTENERS_NOD01 =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST = nod01-
vip) (PORT = 1521)))

NOD01 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = nod01-
vip) (PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP) (HOST = nod02-
vip) (PORT = 1521))
    (LOAD_BALANCE = yes)
    (CONNECT_DATA =
      (SERVICE_NAME = NOD01)
      (FAILOVER_MODE =
        (TYPE = SELECT)
        (METHOD = BASIC)
        (RETRIES = 200)
        (DELAY = 5)
      )
    )
  )
```

Estos archivos, como ya se ha comentado, son indispensables para el buen funcionamiento de Clusterware en la red. Sin una configuración adecuada de ambos el listener es incapaz de levantarse.

Recolocación de los archivos de la base de datos

Ahora vamos a recolocar los archivos de la instancia de la base de datos en nuestra nueva configuración, esto es **pasar de ASM en single-instance a ASM en cluster**.

Apuntar que de todos los archivos que son modificados, eliminados o recreados, se guarda una copia del original en el directorio `/home/oracle/setups/backups` aunque no se muestra en la ejecución.

Para conseguir nuestro objetivo, lo primero que debemos hacer es crear un **nuevo archivo pfile a partir de nuestro actual spfile**, el cual llamaremos `tmppfile.ora` y lo situaremos temporalmente en directorio indicado en el comando:

```
SQL> create pfile='/home/oracle/setups/tmppfile.ora'  
      from spfile;
```

Tras crearlo accedemos a él y **añadimos las siguientes líneas**:

```
[oracle@nod01 setups]$ gedit tmppfile.ora  
  
*.cluster_database = TRUE  
*.cluster_database_instances = 1  
*.undo_management=AUTO  
DBase10.instance_name=DBase10  
DBase10.instance_number=1  
DBase10.thread=1  
DBase10.local_listener=LISTENER_NOD01
```

Como se puede observar, las líneas que hemos añadido son para indicar en el archivo de las características de la base de datos que ahora está en cluster, que es la instancia número 1, su listener, etc.

Una vez hechos los cambios en el `pfile` hay que hacer que sean efectivos en nuestra base de datos. Para ello debemos **recrear el spfile** y por lo tanto debemos parar la base de datos, relevantarla con nuestro nuevo `pfile` y crear nuestro `spfile` a partir del `pfile`:

```
SQL> shutdown immediate  
  
Database closed.  
Database dismounted.  
ORACLE instance shut down.
```

```
SQL> startup pfile=/home/oracle/setups/tmppfile.ora  
ORACLE instance started.  
  
Total System Global Area 276197376 bytes  
Fixed Size 1299372 bytes  
Variable Size 150998100 bytes  
Database Buffers 121634816 bytes  
Redo Buffers 2265088 bytes  
Database mounted.  
Database opened.
```

```
SQL> create spfile='+DATA/dbase10/spfiledbase10.ora'  
      from pfile='/home/oracle/setups/tmppfile.ora';  
  
File created.
```

Tras la creación del nuevo `spfile` debemos **editar el `initDBase10.ora`** de nuestra base de datos con la nueva dirección del `spfile`:

```
[oracle@nod01 setups]$ gedit initDBase10.ora  
  
spfile='+DATA/dbase10/spfiledbase10.ora'
```

Ahora debemos crear un **nuevo archivo de password**. Borramos el actual y lo recreamos utilizando `orapwd`:

```
[oracle@nodol1 dbs]$ rm orapwDBase10
[oracle@nodol1 dbs]$ orapwd file=orapwDBase10
password=oracle
```

Tras todas estas ejecuciones debemos volver a parar la base de datos y volver a levantarla y así comprobar que todo funciona correctamente:

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup
ORACLE instance started.

Total System Global Area 276197376 bytes
Fixed Size 1299372 bytes
Variable Size 150998100 bytes
Database Buffers 121634816 bytes
Redo Buffers 2265088 bytes
Database mounted.
Database opened.
```

Por ultimo en cuanto a esta configuración del spfile se refiere, después de comprobar que arranca correctamente, **comprobamos cuál es el spfile que está utilizando:**

```
SQL> show parameter spfile

NAME                                TYPE
VALUE
-----
spfile                               string
+DATA/dbase10/spfiledbase10.ora
```

Configuraciones para RAC

Ya comprobado que la base de datos arranca con el spfile correcto pasamos a **añadir la instancia de ASM**, ASM1, al CRS. Esto es recomendable para una mayor disponibilidad ya que CRS puede detectar cualquier fallo que se produzca en la instancia e intentar levantarla de nuevo o, en el caso de que se produjera, levantarla automáticamente tras una caída del servidor:

```
[oracle@nodol1]$ srvctl add asm -n nodol1 -i +ASM1 -o
/u01/app/oracle/product/11.1.0/db_1
[oracle@nodol1]$ srvctl start asm -n nodol1
[oracle@nodol1]$ srvctl status asm -n nodol1
ASM instance +ASM1 is running on node nodol1.
```

Tras meter en el CRS la instancia de ASM pasamos a la **creación de las vistas de diccionario de datos de RAC**. Para ello debemos ejecutar el script `catclust` y señalar la salida del script en `backups`, como en otras ocasiones:

```
SQL> spool /home/oracle/setups/backups
SQL> @ $ORACLE_HOME/rdbms/admin/catclust
```

El resultado de la ejecución del script es la siguiente:

```
Package created.

Package body created.
```

PL/SQL procedure successfully completed.

View created.

Synonym created.

Grant succeeded.

View created.

Grant succeeded.

Synonym created.

View created.

Grant succeeded.

Synonym created.

View created.

Grant succeeded.

Synonym created.

View created.

Grant succeeded.

Synonym created.

View created.

Grant succeeded.

Synonym created.

View created.

Grant succeeded.

Synonym created.

View created.

Grant succeeded.

Synonym created.

View created.

Grant succeeded.

View created.

Grant succeeded.

Synonym created.

Synonym created.

Synonym created.

Synonym created.

Synonym created.

Synonym created.

Synonym created.

Synonym created.

Synonym created.

Synonym created.

Synonym created.

Synonym created.

View created.

Grant succeeded.

Synonym created.

View created.

```
Grant succeeded.

Synonym created.

PL/SQL procedure successfully completed.
```

Tras la ejecución de este script y ver que todo ha ido correctamente **cerramos la salida y la base de datos:**

```
SQL> spool off
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

Nuestro siguiente paso es **registrar la instancia de base de datos en el CRS:**

```
[oracle@nod01 ~]$ srvctl add database -d DBase10 -o
$ORACLE_HOME
[oracle@nod01 ~]$ srvctl status database -d DBase10
[oracle@nod01 ~]$ srvctl add instance -d DBase10 -i
DBase10 -n nod01
```

Levantamos la base de datos desde Clusterware:

```
[oracle@nod01 ~]$ srvctl start instance -d DBase10 -i
DBase10
```

Comprobamos el estado de todos los targets que tiene actualmente nuestro Clusterware:

```
[oracle@nod01 bin]$ ./crs_stat -t
```

Name	Type	Target	State	Host
ora....10.inst	application	ONLINE	ONLINE	nod01
ora.DBase10.db	application	ONLINE	ONLINE	nod01
ora....SM1.asm	application	ONLINE	ONLINE	nod01
ora....01.lsnr	application	ONLINE	ONLINE	nod01
ora.nod01.gsd	application	ONLINE	ONLINE	nod01
ora.nod01.ons	application	ONLINE	ONLINE	nod01
ora.nod01.vip	application	ONLINE	ONLINE	nod01

Comprobamos el estado de la base de datos en nuestro RAC:

```
[oracle@nod01 bin]$ srvctl status database -d DBase10
Instance DBase10 is running on node nod01
```

Por ultimo **comprobamos los datos básicos de la base de datos y su estado** consultando gv\$instance:

```
SQL> select instance_number instance#, instance_name,
host_name, status from gv$instance;
INSTANCE# INSTANCE_NAME HOST_NAME STATUS
-----
1 DBase10 nod01.acshost.com OPEN
```

Como se puede observar los datos son correctos y la base de datos está abierta y funcionando correctamente.

Antes de dar por concluido todo el proceso consultamos los datos que creamos al principio sobre la base de datos 10.2.0.4. De esta manera podremos afirmar definitivamente que la migración ha sido un éxito.

Nos logueamos con el usuario que creamos, miguel:

```
SQL> connect miguel
Enter password:
Connected.
SQL> show user
USER is "MIGUEL"
```

Consultamos la tabla que tenía creada:

```
SQL> select * from albums;
NOMALBUM
-----
GRUPO
-----
FLANZAMIENTO NCANCIONES
-----
Night fall in Middle Earth
Blind Guardian
          1998          22
```

Por último, **comprobamos que las características de la base de datos** han cambiado acordes a las configuraciones que hemos hecho:

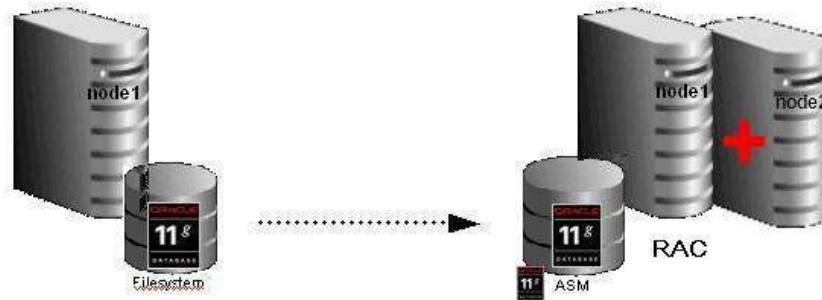
```
SQL> select * from v$instance;
INSTANCE_NUMBER INSTANCE_NAME
-----
HOST_NAME
-----
VERSION          STARTUP_T STATUS          PAR
-----
THREAD# ARCHIVE LOG_SWITCH_WAIT
-----
LOGINS          SHU DATABASE_STATUS  INSTANCE_ROLE
-----
ACTIVE_ST BLO
-----
          1 DBase11
nodol.acshost.com
11.1.0.6.0          08-DEC-09 OPEN          YES
1 STOPPED
ALLOWED          NO ACTIVE          PRIMARY_INSTANCE
NORMAL          NO
```

Tras comprobar que las características de la base de datos son las correctas y que los datos se han mantenido se puede afirmar que la migración de 10.2.0.4 en single instance a 11.1.0.6 en RAC y ASM ha sido todo un éxito.

Así pues, damos por **finalizado el proceso de conversión a RAC de nuestra base de datos y con ello todo el proyecto.**

Adición de un segundo nodo al RAC

En este punto se presenta el proceso de adición de nuevos nodos a un RAC existente. Se presenta de forma separada al todo del proceso de migración ya que **no ha sido posible realizarlo** al no disponer de suficientes recursos. Como se comenta más en profundidad en el siguiente apartado, **Problemas encontrados**, el disponer solamente de una máquina y tener que realizar todo el trabajo sobre una máquina virtual ha sido un hándicap para este objetivo.



Así pues, para que quede constancia se muestra todo el proceso hipotético que debería seguirse. **Se muestra algunas configuraciones realizadas** ya que, como se ha comentado, se intentó completar esta configuración pero no se pudo.

Este proceso consta de 7 pasos:

1. Prerrequisitos y Dependencias
2. Configuración de los Componentes de Red
3. Instalación de Oracle Clusterware
4. Configuración de Oracle Clusterware
5. Instalación del Software de Oracle
6. Adición de la Nueva Instancia

Prerrequisitos y Dependencias

En este apartado poco se va a comentar ya que los prerrequisitos necesarios son los mismos que veíamos en el apartado, **Preparación de la migración**. El nuevo nodo necesita tener suficientes recursos para soportar el software de la base de datos y de Clusterware además de tener todos los paquetes necesarios instalados, las opciones y necesidades de red cubiertas y los usuarios y grupos requeridos configurados.

Configuración de los Componentes de Red

Debemos añadir la referencia al nuevo nodo en todos los nodos de nuestro RAC. Así, el archivo `/etc/hosts` quedaría de la siguiente forma en `node1` y en el que queremos añadir, que llamaremos `node2`:

```
192.168.201.128 node1 node1.acshost.com
192.168.201.138 node1-vip node1-vip.acshost.com
192.168.160.128 node1-priv node1-priv.acshost.com
192.168.201.129 node2 node2.acshost.com
```

```
192.168.201.139 nodo2-vip nodo2-vip.acshost.com
192.168.160.129 nodo2-priv nodo2-priv.acshost.com
```

El hecho de añadir estas referencias de ips a los `/etc/hosts` hace evidente que el nodo2 debe ser configurado dentro de las mismas subredes que nodo1. En nuestro caso, que estamos trabajando con máquinas virtuales, deberíamos **configurarle dos adaptadores de red**, al igual que hicimos con nodo1.

Se debe establecer la equivalencia de usuario con SSH. Cuando se añade nodos al cluster, Oracle copia archivos del nodo donde se realizó originalmente la instalación en el nuevo nodo. Este proceso de copia es realizado utilizando el protocolo `ssh` donde es posible o mediante la copia remota (`rcp`). Para que sea posible la operación de copia es necesario que el usuario `oracle` en el nodo del RAC sea capaz de loguearse en el nuevo nodo sin tener que aportar password.

Para establecer la equivalencia hay que **crear la clave de autenticación**. Con la ejecución que mostramos a continuación se crea esta clave, que es guardada en `home/oracle/.ssh/id_rsa.pub`:

```
[oracle@nodo2 bin]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key
 (/home/oracle/.ssh/id_rsa):
/home/oracle/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
 /home/oracle/.ssh/id_rsa.
Your public key has been saved in
 /home/oracle/.ssh/id_rsa.pub.
The key fingerprint is:
41:48:07:56:40:02:a6:72:ce:77:c4:66:77:7a:be:f6
oracle@nodo2.acshost.com
```

La nueva clave debe ser anexada a `/home/oracle/.ssh/id_dsa/authorized_keys`:

```
[oracle@nodo2 bin]$ cat /home/oracle/.ssh/id_rsa.pub
>> /home/oracle/.ssh/authorized_keys
```

Comprobamos que se puede realizar la conexiones sin problema:

```
[oracle@nodo1 bin]$ ping nodo2-priv
PING nodo2-priv (192.168.160.129) 56(84) bytes of
 data.
64 bytes from nodo2-priv (192.168.160.129):
 icmp_seq=1 ttl=64 time=0.360 ms
[oracle@nodo1 bin]$ ping nodo2
PING nodo2 (192.168.201.129) 56(84) bytes of data.
64 bytes from nodo2 (192.168.201.129): icmp_seq=1
 ttl=64 time=0.782 ms
[oracle@nodo2 ~]$ ping nodo1
PING nodo1 (192.168.201.128) 56(84) bytes of data.
64 bytes from nodo1 (192.168.201.128): icmp_seq=1
 ttl=64 time=2.99 ms
[oracle@nodo2 ~]$ ping nodol-priv
PING nodol-priv (192.168.160.128) 56(84) bytes of
 data.
```

```
64 bytes from nodo1-priv (192.168.160.128):  
icmp_seq=1 ttl=64 time=1.28 ms
```

Ahora **almacenamos las claves públicas para los hosts** mediante ssh-keyscan. Esto se realiza para que todos los nodos tengan la clave de todos los demás nodos del cluster y así no se pida en cada conexión la password. Así pues, ejecutamos:

```
[oracle@nodo2 bin]$ ssh-keyscan -t rsa nodo2 nodo2-  
priv nodo1 nodo1-priv >> ~/.ssh/known_hosts  
  
# nodo2 SSH-2.0-OpenSSH_4.3  
# nodo2-priv SSH-2.0-OpenSSH_4.3  
# nodo1 SSH-2.0-OpenSSH_4.3  
# nodo1-priv SSH-2.0-OpenSSH_4.3
```

Por último **copiamos en el nodo1 el archivo known_hosts de nodo2:**

```
[oracle@nodo1 bin]$ scp nodo2:~/.ssh/known_hosts  
~/.ssh/known_hosts  
  
The authenticity of host 'nodo2 (192.168.201.129)'  
can't be established.  
RSA key fingerprint is  
67:90:e9:49:eb:2c:71:bc:4f:d5:31:c9:e7:37:35:e1.  
Are you sure you want to continue connecting  
(yes/no)? yes  
Warning: Permanently added 'nodo2,192.168.201.129'  
(RSA) to the list of known hosts.  
known_hosts 100%  
2352 2.3KB/s 00:00
```

Instalación de Oracle Clusterware

Oracle Clusterware ya está instalado en el cluster. La tarea que hay que realizar en este caso es **añadir el nuevo nodo a la configuración actual**. Esta tarea se realiza mediante la utilidad de Oracle, **addnode**, que se encuentra en el home de Clusterware en `oui/bin`. Clusterware tiene dos archivos, el OCR y el voting disk, que contienen información sobre el cluster y a las aplicaciones administradas por Clusterware. Estos archivos deben ser actualizados con la información del nuevo nodo.

Así pues, lo único que se debe hacer en este apartado es la **verificación de que el nuevo nodo está preparado** para la instalación. Esto lo realizamos mediante dos chequeos. Para el primer nodo utilizamos `runcluvfy`, en este caso utilizamos `cluvfy` desde el nodo1:

```
[oracle@nodo1 bin]$ ./cluvfy stage -post hwos -n  
nodo1,nodo2  
  
Performing post-checks for hardware and operating  
system setup  
  
Checking node reachability...  
Node reachability check passed from node "nodo1".  
  
Checking user equivalence...  
User equivalence check passed for user "oracle".  
  
Checking node connectivity...  
Node connectivity check passed for subnet  
"192.168.201.0" with node(s) nodo2,nodo1.
```

```
Node connectivity check passed for subnet
"192.168.160.0" with node(s) nodo2,nodo1.
Suitable interfaces for the private interconnect on
subnet "192.168.201.0":
nodo2 eth0:192.168.201.50 eth0:192.168.201.55
nodo1 eth0:192.168.201.10 eth0:192.168.201.15
Suitable interfaces for the private interconnect on
subnet "192.168.160.0":
nodo2 eth1:192.168.160.128
nodo1 eth1:192.168.160.129
Checking shared storage accessibility...
Shared storage check passed on nodes "nodo2,nodo1".
Post-check for hardware and operating system setup
was successful on all the nodes.
```

Realizamos un **segundo chequeo**, ahora para comprobar los servicios del cluster para ambos nodos:

```
[oracle@nodo1 bin]$ ./cluvfy stage -pre crsinst -n
nodo1,nodo2 -r 1lgR1
Performing pre-checks for cluster services setup

Checking node reachability...
Node reachability check passed from node "nodo1".

Checking user equivalence...
User equivalence check passed for user "oracle".

Checking administrative privileges...
User existence check passed for "oracle".
Group existence check passed for "oinstall".
Membership check for user "oracle" in group
"oinstall" [as Primary] passed.

Administrative privileges check passed.

Checking node connectivity...

Node connectivity check passed for subnet
"192.168.201.0" with node(s) nodo1,nodo2.
Node connectivity check passed for subnet
"192.168.160.0" with node(s) nodo1,nodo2.

Interfaces found on subnet "192.168.201.0" that are
likely candidates for a private interconnect:
nodo1 eth0:192.168.201.128
nodo2 eth0:192.168.201.129

Interfaces found on subnet "192.168.160.0" that are
likely candidates for a private interconnect:
nodo1 eth1:192.168.160.128
nodo2 eth1:192.168.160.129

WARNING:
Could not find a suitable set of interfaces for VIPs.

Node connectivity check passed.

Checking system requirements for 'crs'...
Total memory check failed.
Check failed on nodes:
```

```
      nodo1,nodo2
Free disk space check passed.
Swap space check failed.
Check failed on nodes:
      nodo1,nodo2
System architecture check passed.
Kernel version check passed.
Package existence check passed for "make-3.81".
Package existence check passed for "binutils-
2.17.50.0.6".
Package existence check passed for "gcc-4.1.1".
Package existence check passed for "libaio-0.3.106".
Package existence check passed for "libaio-devel-
0.3.106".
Package existence check passed for "libstdc++-4.1.1".
Package existence check passed for "elfutils-libelf-
devel-0.125".
Package existence check passed for "sysstat-7.0.0".
Package existence check passed for "compat-libstdc++-
33-3.2.3".
Package existence check passed for "libgcc-4.1.1".
Package existence check passed for "libstdc++-devel-
4.1.1".
Package existence check passed for "unixODBC-2.2.11".
Package existence check passed for "unixODBC-devel-
2.2.11".
Package existence check passed for "glibc-2.5-12".
Group existence check passed for "dba".
Group existence check passed for "oinstall".
User existence check passed for "nobody".

System requirement failed for 'crs'

Pre-check for cluster services setup was unsuccessful
on all the nodes.
```

En este caso apuntar que fallan los prerrequisitos porque, como ya pasaba en el nodo1, estamos trabajando con máquinas virtuales y estaba previsto que se produjeran fallos en memoria y otros.

Configuración de Oracle Clusterware

Como ya se ha comentado anter, la tarea que se debe realizar es **añadir el nodo2 a la configuración existente**. Para ello disponemos de la utilidad addNode, la cual está localizada en el home de Clusterware en `oui/bin`.

A partir de este momento se realiza ya una guía sin capturas de la ejecución ya que nuestra máquina ya no era capaz de realizar los trabajos necesarios.

El script de addNode se encuentra, como ya hemos comentado, en `/u01/crs/oracle/product/11.1.0/crs/oui/bin/`. Se ejecuta y aparece la utilidad gráfica que va guiando por los pasos necesarios.

En el primer paso se debe **introducir los datos del nodo2, el nombre público, privado y virtual**. Tras introducirlos y comprobar la aplicación que los datos son correctos se **comienza la instalación**.

Una vez que se ha instalado todos los componentes de Clusterware necesarios y se ha copiado todos los archivos requeridos del nodo1 en el nodo2, el instalador demanda la ejecución de 3 scripts:

1. `/u01/app/oracle/orainstRoot.sh` en el nodo2

2. `/u01/crs/oracle/product/11.1.0/crs/install/rootaddnode.sh` en el nodo1. Este script añade la información del nuevo nodo al OCR utilizando `srvctl`.
3. `/u01/crs/oracle/product/11.1.0/crs/root.sh` en el nodo2.

Tras ejecutar estos scripts es recomendable **configurar manualmente mediante el VIP Configuration Assistant, VIPCA, la dirección VIP del nodo2**. Ya hemos indicado en pasos anteriores cual queremos que sea en el `/etc/hosts` pero se debe utilizar esta herramienta ya que también configura GSD y ONS.

Para completar la instalación de Clusterware se crean en `/etc/init.d` los archivos `init.evmd`, `init.cssd`, `init.crsd` y `init.crs`.

El `inittab` del nodo2 es actualizado con las siguientes líneas:

```
x:5:respawn:/etc/X11/prefdm -nodaemon
h1:35:respawn:/etc/init.d/init.evmd run >/dev/null
2>&1 </dev/null
h2:35:respawn:/etc/init.d/init.cssd fatal >/dev/null
2>&1 </dev/null
h3:35:respawn:/etc/init.d/init.crsd run >/dev/null
2>&1 </dev/null
```

Tras realizar todas estas operaciones se vuelve a la pantalla del instalador de Clusterware y se **finaliza la instalación**.

Ahora se podría **comprobar el resultado final**. Mostramos las comprobaciones que se debe hacer con unos resultados hipotéticos en el caso de haberlo podido realizar.

Se comprueba que **Clusterware tiene registrados todos los nodos** mediante el comando `olsnodes`:

```
[oracle@nodo1 oracle]$ olsnodes
nodo1
nodo2
```

Se verifica si los **servicios del cluster están levantados** mediante:

```
[oracle@oradb1 oracle]$ crs_stat -t
```

Name	Type	Target	State	Host
ora.nodo1.gsd	application	ONLINE	ONLINE	nodo1
ora.nodo1.ons	application	ONLINE	ONLINE	nodo1
ora.nodo1.vip	application	ONLINE	ONLINE	nodo1
ora.nodo2.gsd	application	ONLINE	ONLINE	nodo2
ora.nodo2.ons	application	ONLINE	ONLINE	nodo2
ora.nodo2.vip	application	ONLINE	ONLINE	nodo2

Por último se verifica que los **servicios de VIP están configurados a nivel de sistema operativo**. La dirección IP virtual debe estar configurada y añadida a la configuración de red del sistema operativo y los servicios de red estar levantados. Se puede comprobar mediante:

```
[oracle@nodo2 oracle]$ ifconfig -a
eth0 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:60
inet addr:192.168.201.128 Bcast:192.168.201.255
Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fec0:b160/64
Scope:Link
eth0:1 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:60
inet addr:192.168.201.138 Bcast:192.168.201.255
Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500
Metric:1
Interrupt:67 Base address:0x2424
```

```
eth1 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:6A
inet addr:192.168.160.128 Bcast:192.168.160.255
Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fec0:b16a/64
Scope:Link
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
```

Instalación del Software de Oracle

El siguiente paso es la instalación del software en el nuevo nodo. Al igual que pasaba con Clusterware, Oracle proporciona también para este trabajo el script `addNode.sh`, localizado en `$ORACLE_HOME/oui/bin`.

El proceso a seguir es similar al de Clusterware. **Se ejecuta el script `addNode.sh`** y aparecen los datos de los nodos existentes en el cluster. Se debe seleccionar el nodo nuevo y se comienza la instalación.

Una vez se ha copiado el software en el nodo nuevo, el instalador demanda la ejecución del script `/u01/app/oracle/product/11.1.0/db_1/root.sh` en el nodo que hemos añadido.

Tras la ejecución de este último script finaliza la instalación. **Se recomienda que se ejecute `netca`** antes de continuar con el proceso y así se configurará los archivos de red y los parámetros como el listener, `sql*net`, y el `tnsnames.ora`.

Adición de la Nueva Instancia

El último paso es la adición de la instancia. Para realizar esto se necesita el DBCA y que Clusterware esté levantado en todos los nodos. Además se recomienda hacer un backup completo en frío de la base de datos antes de comenzar el proceso.

Así pues, se debe **ejecutar DBCA** y elegir la opción *Instance Management* para luego seleccionar *Add an instance*. A partir de este momento DBCA lleva por las diferentes configuraciones que se deben realizar como el **nombre de la instancia o la configuración de los servicios**.

DBCA verifica la configuración del nuevo nodo y, como nuestra base de datos está configurada con ASM, muestra el mensaje *"ASM is present on the cluster but needs to be extended to the following nodes: [oradb5]. Do you want ASM to be extended?"*. Ante esto debe clicarse en *Yes* y **se añade ASM a la nueva instancia**.

Para crear y arrancar la instancia de ASM en el nuevo nodo, Oracle requiere que haya un listener presente y arrancado. Por esto, DBCA pregunta si se quiere **configurar un listener**. Si el puerto por defecto es bueno se clicarse en *Yes* y DBCA realiza el trabajo, si no es así se debe clicarse en *No* y hacerlo manualmente mediante `netca`.

Tras realizar esta última configuración **DBCA termina su trabajo** y, llegados a este punto se obtiene que:

- a) Clusterware está instalado en el nuevo nodo y ahora éste es parte del cluster.
- b) El software de Oracle está instalado en el nuevo nodo.
- c) Las instancias de ASM y base de datos están creadas y configuradas correctamente en el nuevo nodo.

Así pues, se recomienda realizar las comprobaciones oportunas de que el proceso se ha terminado con éxito.

Al igual que hacíamos antes, mostramos los resultados que normalmente obtendríamos de haber podido realizar estos procesos:

Verificamos que **todas las instancias en el cluster están levantadas** mediante la consulta de la vista V\$ACTIVE_INSTANCES en cualquiera de las instancias:

```
SQL> select * from v$active_instances;
INST_NUMBER INST_NAME
-----
1 nod01.acshots.com:DBase101
2 nod02.acshost.com:DBase102
```

Verificamos que **todos los diskgroups de ASM están montados y que los datafiles son visible para la nueva instancia.**

```
SQL> select name, state, type from V$ASM_DISKGROUP;
NAME STATE TYPE
-----
DATA CONNECTED NORMAL
FRA CONNECTED NORMAL
```

```
SQL> select name from V$DATAFILE;
```

```
NAME
-----
+DATA/dbase10/datafile/system.256.701977649
+DATA/dbase10/datafile/undotbs1.258.701977851
+DATA/dbase10/datafile/sysaux.257.701977775
+DATA/dbase10/datafile/users.260.701977861
```

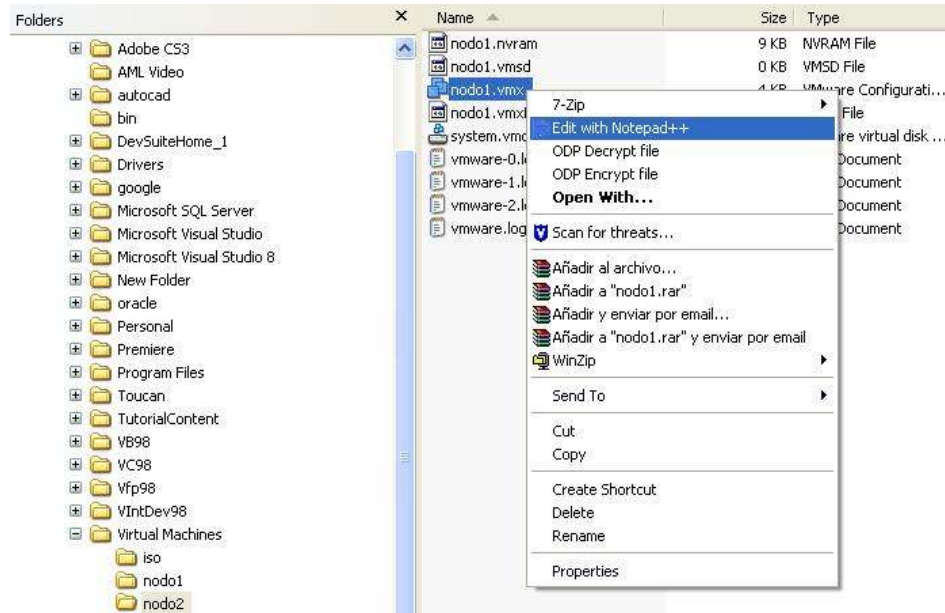
Verificamos que el **OCR es consciente de la nueva instancia en el cluster:**

```
[oracle@nod01 oracle]$ srvctl status database -d
Dbase10
Instance DBase101 is running on node nod01
Instance DBase102 is running on node nod02
```

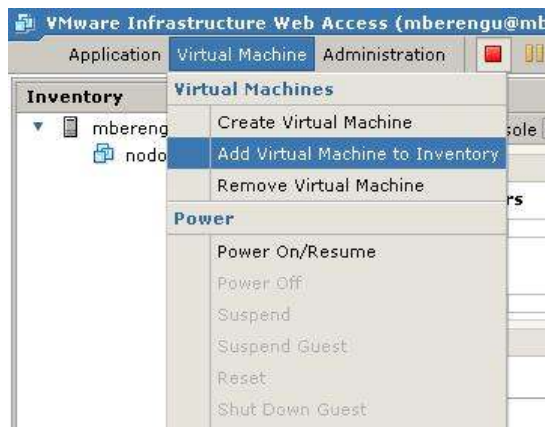
Verificamos que el **OCR es consciente de los servicios de la base de datos:**

```
[oracle@nod01 oracle]$ srvctl status service -d
Dbase10
Service CRM is running on instance(s) DBase101
Service CRM is running on instance(s) Dbase102
Service PAYROLL is running on instance(s) DBase101
Service PAYROLL is running on instance(s) DBase102
```

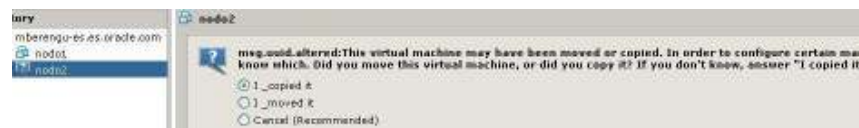
Tras comprobar que todo está correcto **se puede dar por concluido el proceso.** En este momento se dispondría de un RAC con dos nodos.



El siguiente paso fue añadir la máquina a nuestro servidor VMware. Para ello, en la consola de administración seleccionamos *Virtual Machine* y luego *Add Virtual Machine to Inventory*:



Una vez añadida, cuando se arrancaba por primera vez, simplemente había que indicar que había sido copiada y ya se tenía configuradas las máquinas virtuales listas para iniciar el trabajo:



Esta solución funcionó relativamente bien mientras las máquinas no tuvieron que soportar mucha carga. Una vez configurada y trabajando la base de datos 10.2.0.4 en el primer nodo, solamente instalar Clusterware hacía que se ralentizara las máquinas de una forma desmesurada incluso llegando a caer las mismas dos veces durante esta instalación, teniendo que limpiar el fallo y volviendo a instalar.

Tras comprobar que era inviable mantener el sistema configurado de esta forma **se decidió trabajar con un solo nodo**.

2. La segunda opción que se intentó fue hacer **todo el proyecto con un solo nodo** y, tras hacer toda la migración y el paso a RAC, **añadir el segundo nodo**. **Esta es la solución que se ha implementado** aunque, como se ha comentado a lo largo del proyecto tampoco se pudo realizar por la misma razón de falta de recursos.

Fallo en la configuración del virtual host

Tras la instalación de clusterware nos encontramos con un problema con el que no se contaba: **la dirección vip y el listener del cluster no se levantaban**. El instalador había fallado al configurar la vip mostrando los errores CRS-1006: No more members to consider y CRS-0215: Could not start resource 'ora.nodol.vip. Esto hacía que el cluster no funcionara correctamente. **El error fue corregido tras la migración** ya que se prefirió continuar con lo demás antes de acometer este problema ya que esto no impedía el que se realizara la migración.

Si nos logueábamos como root y consultábamos el **estado del cluster** aparecía lo siguiente:

```
[root@nodol bin]# srvctl status nodeapps -n nodol
VIP is not running on node: nodol
GSD is running on node: nodol
Listener is not running on node: nodol
ONS daemon is running on node: nodol
```

Igualmente, si comprobábamos el **estado de todos los targets** aparecía:

```
[root@nodol bin]# crs_stat -t
Name                Type                Target              State              Host
-----
ora....11.inst      application         ONLINE             ONLINE             nodol
ora.DBase11.db      application         ONLINE             ONLINE             nodol
ora....SM1.asm      application         ONLINE             ONLINE             nodol
ora....O1.lsnr      application         ONLINE             OFFLINE           nodol
ora.nodol.gsd       application         ONLINE             ONLINE             nodol
ora.nodol.ons       application         ONLINE             ONLINE             nodol
ora.nodol.vip       application         ONLINE             OFFLINE           nodol
```

Así pues, para conocer si había habido algún error en la configuración se fueron consultando diferentes parámetros:

```
[root@nodol bin]# ./oifcfg getif
eth0 192.168.201.0 global public
eth1 192.168.160.0 global cluster_interconnect
```

Ahora comprobamos que existían el host virtual, GSD, ONS y el listener en el cluster, por si hubiera pasado algo más anormal y ni ya ni siquiera existieran. Como se puede ver **la ip que nos devolvía era precisamente la correcta**:

```
[root@nodol racg]# srvctl config nodeapps -n nodol -a
-g -s -l
VIP exists.: /nodol-vip/192.168.201.138/
255.255.255.0/eth0
GSD exists.
ONS daemon exists.
Listener exists.
```

Al no estar levantado el listener del cluster intentamos **levantarlo a mano mediante lsnrctl**, lo cual devolvía lo siguiente:

```
[oracle@nodol ~]$ lsnrctl start LISTENER_NODOL
```

```
LSNRCTL for Linux: Version 11.1.0.6.0 - Production on
11-JAN-2010 14:38:40
Copyright (c) 1991, 2007, Oracle. All rights
reserved.
Starting
/u01/app/oracle/product/11.1.0/db_1/bin/tnslsnr:
please wait...
TNSLSNR for Linux: Version 11.1.0.6.0 - Production
System parameter file is
/u01/app/oracle/product/11.1.0/db_1/network/admin/li
stener.ora
Log messages written to
/u01/app/oracle/product/11.1.0/db_1/log/diag/tnslsnr
/nod01/listener_nod01/alert/log.xml
Error listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=nod01-
vip)(PORT=1521)(IP=FIRST)))
TNS-12545: Connect failed because target host or
object does not exist
TNS-12560: TNS:protocol adapter error
TNS-00515: Connect failed because target host or
object does not exist
Linux Error: 99: Cannot assign requested address
Listener failed to start. See the error message(s)
above...
```

Tras comprobar que el listener no era capaz de arrancar intentamos parar todo el cluster y rearrancarlo todo de nuevo. Para ello tratamos de **parar el cluster**:

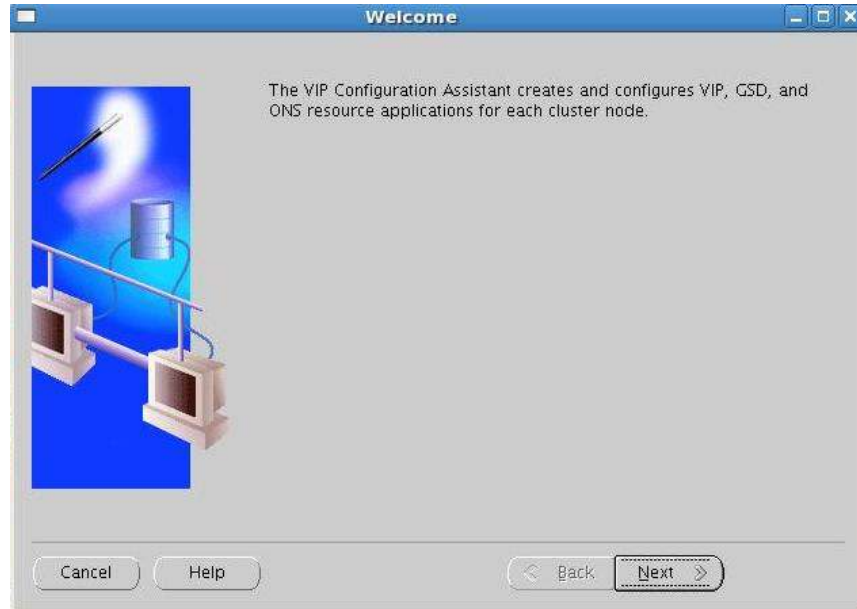
```
[root@nod01 bin]# srvctl stop nodeapps -n nod01
CRS-0216: Could not stop resource
'ora.nod01.LISTENER_NOD01.lsnr'.
```

Con el mensaje de error mostrado comprobamos que el listener era inaccesible, no se podía ni arrancar ni parar. Igualmente intentamos **levantar de nuevo el cluster**:

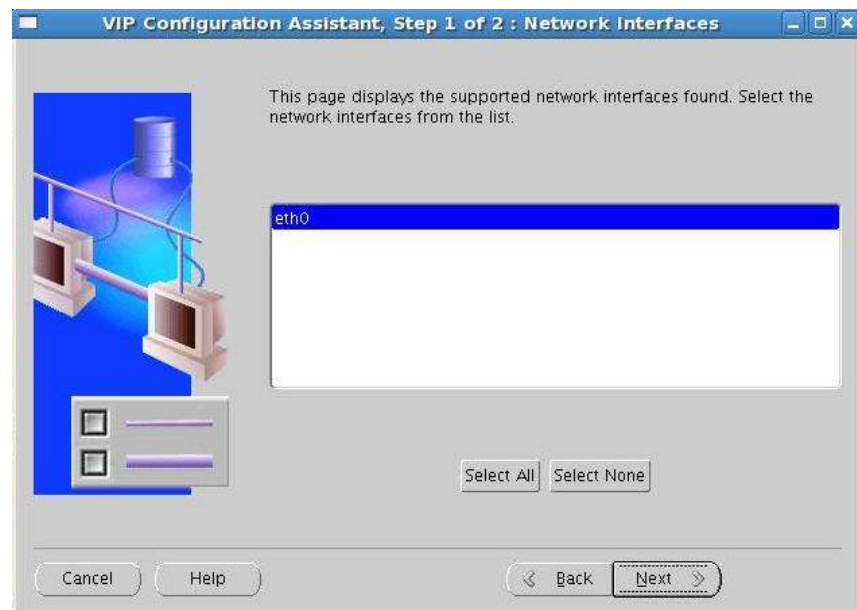
```
[root@nod01 bin]# srvctl start nodeapps -n nod01
nod01:ora.nod01.vip:ping to 192.168.160.2 via eth0
failed, rc = 1 (host=nod01.acshost.com)
nod01:ora.nod01.vip:ping to 192.168.160.2 via eth0
failed, rc = 1 (host=nod01.acshost.com)
nod01:ora.nod01.vip:Interface eth0 checked failed
(host=nod01.acshost.com)
nod01:ora.nod01.vip:Invalid parameters, or failed to
bring up VIP (host=nod01.acshost.com)
CRS-1006: No more members to consider
CRS-0215: Could not start resource 'ora.nod01.vip'.
nod01:ora.nod01.vip:ping to 192.168.160.2 via eth0
failed, rc = 1 (host=nod01.acshost.com)
nod01:ora.nod01.vip:ping to 192.168.160.2 via eth0
failed, rc = 1 (host=nod01.acshost.com)
nod01:ora.nod01.vip:Interface eth0 checked failed
(host=nod01.acshost.com)
nod01:ora.nod01.vip:Invalid parameters, or failed to
bring up VIP (host=nod01.acshost.com)
CRS-1006: No more members to consider
CRS-0215: Could not start resource
'ora.nod01.LISTENER_NOD01.lsnr'.
```

Comprobamos que ahora aparecían los errores CRS-1006 y CRS-0215. Los mismos que nos aparecieron en la instalación.

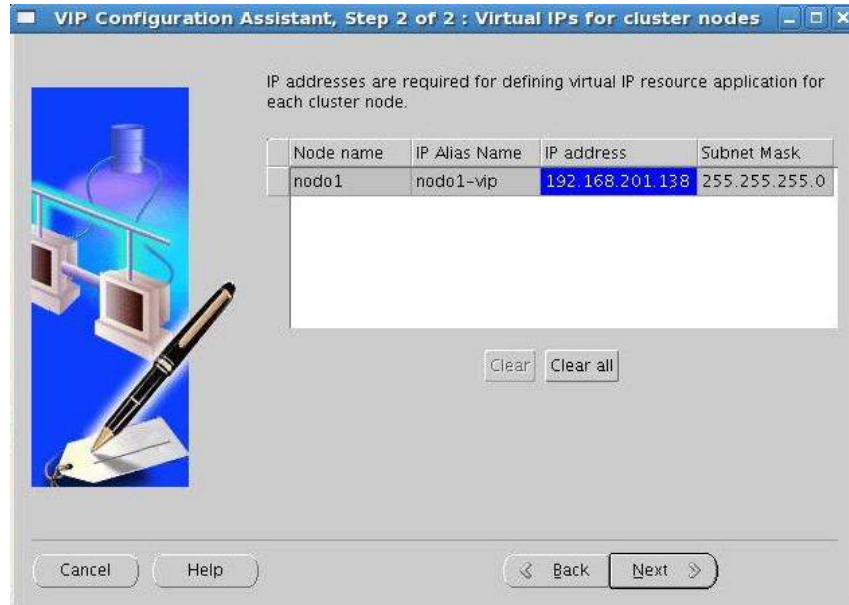
Al ver que VIPCA, no había configurado la dirección vip y entendiendo que el listener no se levantaba al no estar ésta disponible, pasamos a **configurar manualmente esta dirección** mediante este programa.



Tras clicar en la pantalla de bienvenida en *Next* llegamos a la segunda donde se nos muestra las interfaces de red encontradas y soportadas. En este caso está eth0, que es la que queremos:



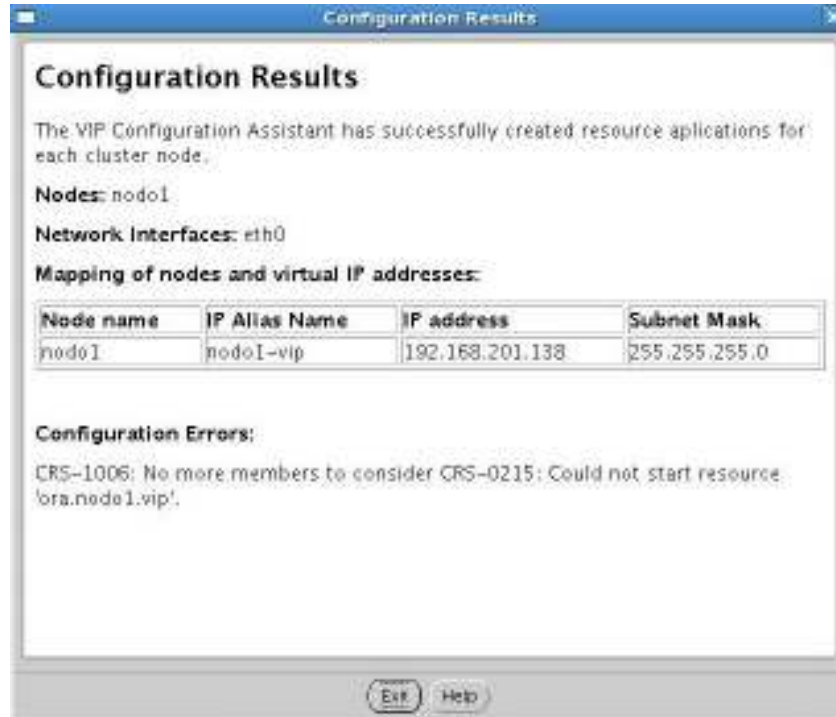
En la siguiente pantalla se nos muestra las características que va a tener la dirección vip. Podemos editarlas pero las que se nos muestran son las correctas, la ip virtual debe ser 192.128.201.138. Así pues, continuamos.



En la siguiente pantalla se nos muestra las características del nodo, interfaz de red, alias,... con las que se va a configurar la dirección vip. Clicamos en *Finish* y empieza el proceso de configuración:



Tras realizarse todo el proceso se nos muestra la pantalla siguiente en la cual aparecen de nuevo los mismos errores que ya aparecieron en la instalación:



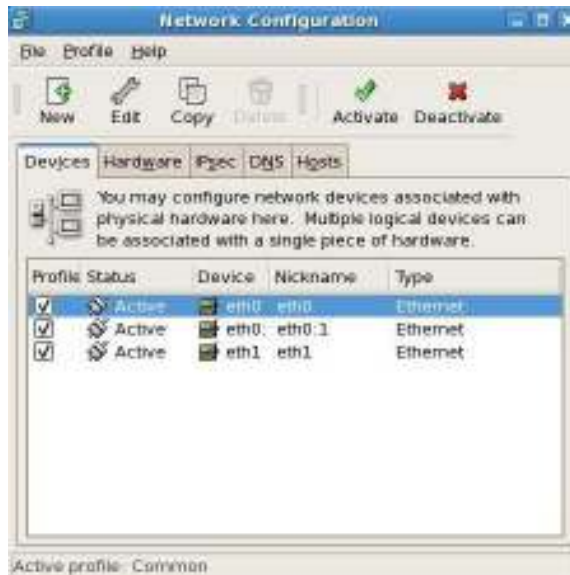
Después de comprobar que no había funcionado esto decidimos **comprobar el estado de las interfaces de red:**

```
[oracle@nod01 ~]$ ifconfig | grep net
eth0 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:60
      inet addr:192.168.201.128 Bcast:192.168.201.255
      Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:fec0:b160/64
      Scope:Link
eth1 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:6A
      inet addr:192.168.160.128 Bcast:192.168.160.255
      Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:fec0:b16a/64
      Scope:Link
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
```

Como era de esperar la **interfaz relativa a la ip virtual no estaba levantada**, ya que debería ser **eth0:1**. Así pues procedimos a reconfigurarla mediante el interfaz gráfico disponible:



La configuración final de las interfaces de red es:



Ahora podíamos comprobar, tras reiniciar los servicios de red, que las interfaces de red estaban configuradas correctamente:

```
[oracle@nodol ~]$ ifconfig | grep net
eth0 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:60
      inet addr:192.168.201.128 Bcast:192.168.201.255
      Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:fec0:b160/64
      Scope:Link
eth0:1 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:60
      inet addr:192.168.201.138 Bcast:192.168.201.255
      Mask:255.255.255.0
eth1 Link encap:Ethernet HWaddr 00:0C:29:C0:B1:6A
      inet addr:192.168.160.128 Bcast:192.168.160.255
      Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:fec0:b16a/64
      Scope:Link
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
```


Por si hubiera habido algún problema **reejecutamos vipca** para volver a asignar la vip que deseamos. El proceso es exactamente igual al visto antes salvo que ahora no se produjo ningún error y la pantalla que se nos mostró al final es la siguiente:



Tras esto procedimos a **levantar el listener** mediante:

```
[oracle@node1 ~]$ lsnrctl start LISTENER_NODO1
LSNRCTL for Linux: Version 11.1.0.6.0 - Production on
11-JAN-2010 14:39:26

Copyright (c) 1991, 2007, Oracle. All rights
reserved.

Starting
/u01/app/oracle/product/11.1.0/db_1/bin/tnslsnr:
please wait...

TNSLSNR for Linux: Version 11.1.0.6.0 - Production
System parameter file is
/u01/app/oracle/product/11.1.0/db_1/network/admi
n/listener.ora
Log messages written to
/u01/app/oracle/product/11.1.0/db_1/log/diag/tns
lsnr/node1/listener_node1/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.16
8.201.138)(PORT=1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.16
8.201.128)(PORT=1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC
)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=node1-
vip)(PORT=1521)(IP=FIRST)))
STATUS of the LISTENER
```

```
-----  
Alias                LISTENER_NOD01  
Version              TNSLSNR for Linux: Version  
                    11.1.0.6.0 - Production  
Start Date           11-JAN-2010 14:39:26  
Uptime               0 days 0 hr. 0 min. 0 sec  
Trace Level          off  
Security             ON: Local OS Authentication  
SNMP                 OFF  
Listener Parameter File  
                    /u01/app/oracle/product/11.1.0/db_1/network/admi  
                    n/listener.ora  
Listener Log File  
                    /u01/app/oracle/product/11.1.0/db_1/log/diag/tns  
                    lsnr/nod01/listener_nod01/alert/log.xml  
Listening Endpoints Summary...  
  
    (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.16  
    8.201.138)(PORT=1521)))  
  
    (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.16  
    8.201.128)(PORT=1521)))  
    (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC)))  
The listener supports no services  
The command completed successfully
```

Por ultimo levantamos de nuevo todo el cluster:

```
[root@nod01 bin]# srvctl start nodeapps -n nod01
```

La ejecución de `srvctl start` **no produjo ningún error**, por lo que se entendía que todo había funcionado ya correctamente y el cluster debía estar ejecutándose correctamente. Esto lo pudimos comprobar mediante las siguientes ejecuciones:

```
[root@nod01 bin]# srvctl status nodeapps -n nod01
```

```
VIP is not running on node: nod01  
GSD is running on node: nod01  
Listener is not running on node: nod01  
ONS daemon is running on node: nod01
```

```
[root@nod01 bin]# crs_stat -t
```

Name	Type	Target	State	Host
ora....11.inst	application	ONLINE	ONLINE	nod01
ora.DBBase11.db	application	ONLINE	ONLINE	nod01
ora....SM1.asm	application	ONLINE	ONLINE	nod01
ora....O1.lsnr	application	ONLINE	ONLINE	nod01
ora.nod01.gsd	application	ONLINE	ONLINE	nod01
ora.nod01.ons	application	ONLINE	ONLINE	nod01
ora.nod01.vip	application	ONLINE	ONLINE	nod01

Así pues, finalmente **el cluster estaba ya configurado y ejecutándose de manera correcta**.

Configuración de ASM en single instance

En un primer momento y para “complicar” un tanto más el proyecto se pensó en realizar la migración de la base de datos de 10.2.0.4 a 11.1.0.6 y de file system a ASM pero, en vez de cómo se ha hecho, directamente a ASM en cluster, primero pasar a ASM en single instance y luego, con el paso a RAC también cambiar ASM en cluster.

Para realizar esta configuración, en el paso **Instalación de ASM**, se realizó la misma en single instance configurando los mismos diskgroups. La migración a ASM se realizó tal y como hemos indicado a lo largo del proyecto. Sin embargo, una vez configurada la base de datos 11.1.0.6 en single instance con ASM en single instance también se realizó otra migración distinta a RAC.

Hay que recordar que la forma en que hemos realizado la configuración de ASM es directamente en cluster. Finalmente lo hemos mostrado así en el proyecto ya que es la forma más intuitiva y, a nuestro parecer, la más recomendable en una migración de este tipo.

La otra opción de **migración de ASM** que realizamos fue **mediante DBCA**. Mediante esta herramienta la migración es sencilla. Ya que no se dispone de capturas, se puede decir resumiendo que sólo tuvimos que quitar la línea referente al autoarranque de CSS en el archivo /etc/inittab y ejecutar:

```
[oracle@nodol bin]$ dbca -silent -deleteASM
Look at the log file
"/u01/app/oracle/cfgtoollogs/dbca/silent.log" for
further details.
```

Esto permite desinstalar la instancia de ASM sin perder los datos.

Tras esto y, teniendo ya instalado Clusterware, se ejecutó DBCA y se le indicó para configurar ASM en cluster, igual que se muestra en el apartado **Instalación de ASM**. Cuando aparecieron los diskgroups disponibles, simplemente seleccionando los ya configurados **se configuraba la nueva instancia en cluster con los mismos datos**.

Migración a RAC mediante RCONFIG

La migración a RAC se puede hacer de distintas formas. **Se puede hacer mediante DBCA, Enterprise Manager, rconfig o manualmente**, como hemos expuesto en el desarrollo de este proyecto.

A diferencia de la razón por la que se ha elegido la forma más sencilla de configurar ASM, para realizar la migración a RAC se utilizó la más laboriosa. Los otros métodos son más automáticos y, a priori, más fáciles de ejecutar. En nuestro caso se ha realizado de esta manera para obtener un conocimiento mayor y mostrar de una manera más extensa todo el proceso y complejidad que representa una migración de este tipo.

Sin embargo, al igual que ha pasado con ASM, también se probó la migración mediante otro sistema al explicado en el proyecto, **rconfig**. Esto se hizo sobre la base de datos ya con ASM en cluster.

Para ejecutar este proceso se tuvo que **editar el archivo ConvertToRAC.xml**, situado en \$ORACLE_HOME/assistants/rconfig/sampleXMLs. Con la ayuda de los comentarios que aparecen en este archivo se introdujeron los datos acordes a nuestro sistema.

Una vez editado ConvertToRAC.xml, se ejecutó lo siguiente:

```
[oracle@nodol sampleXMLs]$ rconfig ConvertToRAC.xml
Converting Database DBase10.nodol.acshost.com to
Cluster Database. Target Oracle Home :
/u01/app/oracle/product/11.1.0/db_1.
Setting Data Files and Control Files
Adding Database Instances
Adding Redo Logs
```

```
Enabling threads for all Database Instances
Setting TEMP tablespace
Adding UNDO tablespaces
Adding Trace files
Setting Flash Recovery Area
Updating Oratab
Creating Password file(s)
Configuring Listeners
Configuring related CRS resources
Adding NetService entries
Starting Cluster Database
Starting Listeners
<?xml version="1.0" ?>
<RConfig version="1.1" >
<ConvertToRAC>
  <Convert>
    <Response>
      <Result code="1" >
        Got Exception
      </Result>
    </Response>
  </Convert>
</ConvertToRAC>

oracle.sysman.assistants.rconfig.engine.ListenerStar
tupException
Operation Failed. Refer logs at
/u01/app/oracle/cfgtoollogs/rconfig/rconfig_11_21_09
_18_50_55.log for more details.

  </ErrorDetails>
</Response>
</Convert>
</ConvertToRAC></RConfig>
```

Como se puede observar, la ejecución tuvo un **fallo en el arranque del listener**. Este fallo vino **producido por el error antes comentado de la no configuración de la dirección vip** y, por lo tanto, de la incapacidad para levantar el listener del cluster.

El hecho de que fallara la ejecución tuvo como consecuencia el que se tuviera que echar mano de un backup del sistema para volver a probar la ejecución.

Una vez solucionado el problema como se muestra en el apartado anterior se pudo **realizar la migración sin problemas**.

Múltiples

A lo largo de todo el proyecto han aparecido muchos errores, en muchos casos provocados por mí al no conocer bien el software o la ejecución que debía hacer. En la mayoría de casos se solventaba volviendo a configurar todo, reejecutando o empezando desde el principio de nuevo.

El primer error de importancia que se produjo fue provocado, y no he conseguido saber por qué, por el propio servidor de VMware. Un día, por alguna razón, **dejó de funcionar la primera red virtual** y con ella eth0 y mi posibilidad de transferirle archivos a la máquina virtual ya que lo tenía que hacer mediante ftp. No hubo forma de reconfigurarla ni crear una nueva que funcionara. Así pues, no quedó más alternativa que volver a crear la máquina virtual.

Otro error de gravedad fue la **corrupción de los discos virtuales de la máquina virtual**, ya con el proyecto avanzado, intentando configurar el segundo nodo para agregarlo al RAC. Para generar el segundo nodo se puede hacer de dos formas,

haciendo una *clonación* del nodo existente o creando uno desde 0. Evidentemente es menos costoso *clonar* el nodo.

Se creó el nuevo nodo a partir del primero pero, para realizar todo el proceso de adición de nodo tal cual se ha explicado anteriormente, se tuvo que eliminar cosas como la base de datos y desconfigurar otras como el clusterware. Esto dio como resultado que, al utilizar ambos nodos el mismo almacenamiento en el servidor de VMware, al volver a arrancar el primer nodo, algunas configuraciones ya no funcionaban como correspondía, como era el caso de Clusterware, el cual no arrancaba y devolvía el error CRS-0184: Cannot communicate with the CRS daemon. Viendo que todo había sido trastocado, se tuvo que utilizar un backup y configurar el segundo nodo desde cero, aunque como ya se ha comentado antes, finalmente no se pudo utilizar.

Durante el paso a ASM y especialmente a RAC se realizaron diferentes pruebas. Como se ha mostrado antes, una de ellas fue, por ejemplo, el paso a RAC mediante RCONFIG. Como resultado de estas pruebas y algunos errores que se produjeron en las mismas **se configuraron varios listeners**. Estos listeners, por alguna razón, quedaban configurados aún habiendo sido eliminados mediante netca llegando a darse la siguiente salida:

```
[oracle@nod01 bin]$ ./crs_stat -t
```

Name	Type	Target	State	Host
ora....10.inst	application	ONLINE	ONLINE	nod01
ora.DBase10.db	application	ONLINE	ONLINE	nod01
ora....SM1.asm	application	ONLINE	ONLINE	nod01
ora....01.lsnr	application	ONLINE	OFFLINE	
ora....01.lsnr	application	ONLINE	OFFLINE	
ora....01.lsnr	application	ONLINE	ONLINE	nod01
ora....01.lsnr	application	ONLINE	OFFLINE	
ora.nod01.gsd	application	ONLINE	ONLINE	nod01
ora.nod01.ons	application	ONLINE	ONLINE	nod01
ora.nod01.vip	application	ONLINE	ONLINE	nod01

Además de estos listeners se produjo un error que también impedía el trabajo correcto del sistema, el **tnsnames.ora desapareció**. Así pues, la solución a esto fue la edición del listener.ora y de tnsnames.ora a mano, tal y como se muestra en la página 150 y la eliminación de los listeners mediante srvctl del cluster:

```
[oracle@nod01 bin]$ srvctl remove listener -n nod01 -  
1 LISTENER_RCONFIG_NOD01
```

Conclusiones

La migración de un sistema de base de datos, tras la realización de todo el proyecto, es un trabajo costoso. Realizar una migración es un trabajo laborioso más que por el proceso en sí, por la cantidad de comprobaciones, modificaciones y detalles que hay que tener en cuenta a la hora de acometerlo.

En el caso de este proyecto, se ha realizado una **migración de una base de datos** 10.2.0.4 a una 11.1.0.6 siendo la base de datos lo más simple y básica posible (solamente se le añadió un usuario y una tabla nuevas). El hecho de realizar la migración de una base de datos en un sistema real llevaría todavía más estudio y pruebas ya que los sistemas de producción suelen tener aplicaciones, que también se deberían migrar, y no se puede permitir la caída del servicio durante mucho tiempo o, directamente, no se puede permitir la caída por lo que se tiene que echar mano de procesos auxiliares como la creación de una standby, clonar, realizar un cambio para que el servicio lo de otra base de datos idéntica mientras ésta se está migrando,...

Además de la migración en sí también se ha investigado y realizado un **cambio de almacenamiento**, de file system a ASM, y un **cambio a RAC o clusterización** de la base de datos. Estos cambios, al igual que la migración de la base de datos, conllevan un tiempo de estudio y preparación elevados y se debe tener en cuenta muchos aspectos.

El tiempo invertido en la elaboración del proyecto ha sido elevado. Sin embargo, teniendo en cuenta el documento de Oracle *ds-dboptions_RAC_ACS_2009* que indica que los técnicos de Oracle ACS son capaces de instalar un RAC de dos nodos, previa preparación de las máquinas por parte del cliente, en siete días; se considera que no teniendo el nivel de un técnico de ACS y teniendo un entorno de test, haber realizado en este tiempo la configuración de las máquinas virtuales y preparación para el software Oracle, instalación de base de datos 10g y 11g, instalación de Clusterware y ASM, migración de base de datos de 10g a 11g, cambio de almacenamiento a ASM y clusterización de la base de datos más la documentación de todos los procesos, **es un trabajo estimable**.

Entrando ya en la valoración propia del trabajo, resumiría el proyecto en tres palabras: **conocimiento, experiencia y trabajo**.

El conocimiento es básico y ha sido básico a la hora de desarrollar el proyecto. Actualmente considero que he crecido mucho en cuanto a conocimientos sobre los productos con los que he tenido que trabajar. Un buen trabajo es imposible si no se conoce bien con qué se está trabajando y, en muchas ocasiones, mi avance ha sido más lento del deseado por el hecho de tener que aprender cosas nuevas antes de poder avanzar.

El hecho de no haber trabajado nunca con VMware server, no haber configurado nunca un cluster ni ASM, no haber utilizado nunca RMAN, RCONFIG, VIPCA, Pre_upgrade Information Tool, o simplemente, no haber hecho nunca una migración de base de datos hizo que me enfrentara con un reto y que para superarlo tuviera que documentarme mucho y **adquirir un conocimiento que considero importante**.

Un ejemplo que ilustra muy bien este concepto fue cuando ejecuté Pre- Upgrade Information Tool (página 100). Ante los warnings que aparecían tuve que investigar y algunos los resolví solo. Sin embargo, algunos hacían referencia a parámetros de la base de datos. Debía cambiarlos pero no sabía como. Se lo pregunté a mi jefe y me dijo "créate un pfile a partir del spfile, los cambias y luego sustituyes uno por otro". Esa era la solución. Sin embargo, no tenía ni idea de cómo hacerlo. Con investigación y pruebas acabó saliendo. **Con el conocimiento adecuado todo es más sencillo y rápido**.

La **experiencia es también otro aspecto que considero me ha aportado este proyecto**. Está relacionada directamente con el conocimiento. Al no tener el conocimiento adecuado en muchas ocasiones tuve que echar mano del *prueba y error* encontrando a veces la solución de esta forma. Y muchas veces la mejor forma de aprender es *luchando* con las cosas.

En ocasiones se puede ver un manual y ver que todo debe ir perfectamente, sin embargo la realidad no suele ser así y, el hecho de encontrarte con eso y salvar los obstáculos te confiere un valor añadido, algo que, cuando te vuelva a ocurrir algo semejante, puedes asociar y resolver con mayor facilidad o simplemente, no caer en el mismo error dos veces. Como se suele decir "la experiencia es un grado".

Por último destaco el **trabajo**. La elaboración del proyecto ha sido muy larga. Desde los comienzos en los que buscaba información sobre los motivos para migrar una base de datos y formas de migrar hasta la redacción de los problemas y las conclusiones han pasado meses. Meses en los que se ha tenido que compaginar este proyecto con una asignatura en septiembre y con trabajar por las mañanas, y luego con trabajar mañana y tarde.

La búsqueda de información para la ejecución del proyecto, la planificación del mismo, pruebas, ejecuciones, redacción, reescrituras... todo ha sido un largo proceso que demuestra que realizar algo con pretensiones de ser bueno exige trabajo y sacrificio. Como decía Hector Tassinari "**El único lugar donde encontrarás éxito antes que trabajo es en el diccionario**" y, en mi modesta opinión, este proyecto me lo ha demostrado.

Bibliografía

- Oracle Database Installation Guide 10g Release 2 (10.2) for Linux x86
- Oracle Database Installation Guide 11g Release 1 (11.1) for Linux
- Oracle Database Administrator's Guide 11g Release 1 (11.1)
- Oracle Database Upgrade Guide 11g Release 1 (11.1)
- Oracle Database 11gR1 Upgrade Companion (Version 1.40)
- Upgrade to Oracle Database 11g: Single Instance to RAC & ASM - eSeminar
- Oracle Clusterware Installation Guide 11g Release 1 (11.1) for Linux
- Oracle Real Application Clusters Installation Guide 11g Release 1 (11.1) for Linux and UNIX
- Oracle Clusterware Administration and Deployment Guide 11g Release 1 (11.1)
- RAC Attack Lab Handbook
- Oracle Database 11g_New features_V01
- Convert a Single Instance Database to Oracle RAC 10g on RHEL3
- Nota: 429825.1 - Complete Checklist for Manual Upgrades to 11gR1
- Nota: 283684.1 - How to Change Interconnect/Public Interface IP or Subnet in Oracle
- Nota: 276434.1: - Modifying the VIP or VIP Hostname of a 10g or 11g Oracle Clusterware Node
- Nota: 452758.1 - How to Convert a Single-Instance ASM to Cluster ASM
- Nota: 252219.1 - Steps To Migrate/Move a Database From Non-ASM to ASM And Vice-Versa
- Nota: 726833.1 - Hangcheck-Timer Module Requirements for Oracle 9i, 10g, and 11g RAC on Linux
- Nota: 559365.1 - Using Diagwait as a diagnostic to get more information for diagnosing Oracle Clusterware Node evictions
- Nota: 284752.1 - 10g RAC: Steps To Increase CSS Misscount, Reboottime and Disktimeout
- Nota: 797750.1 - "WARNING: --> Database contains schemas with objects dependent on network packages" When Running utlu111i.sql
- Nota: 556610.1 - Script to Collect DB Upgrade/Migrate Diagnostic Information (dbupgdiag.sql)
- Nota: 560336.1- Script to Check Schemas with Stale Statistics
- Adding New Nodes to Your Oracle RAC 10g Cluster on Linux
- <http://forums.oracle.com/>
- <http://www.orafaq.com>
- ds-dboptions_RAC_ACS_2009
- Oracle Database Concepts
- Oracle Database 2 Day DBA 10g Release 2 (10.2)

- SQL*Plus User's Guide and Reference Release 10.1
- Oracle Database Error Messages 10g Release 2 (10.2)