

StorageTek Enterprise Library Software

Guía de gestión de datos fuera del sitio y recuperación ante desastres

Versión 7.3

E63459-02

Septiembre de 2016

StorageTek Enterprise Library Software

Guía de gestión de datos fuera del sitio y recuperación ante desastres

E63459-02

Copyright © 2015, 2016, Oracle y/o sus filiales. Todos los derechos reservados.

Este software y la documentación relacionada están sujetos a un contrato de licencia que incluye restricciones de uso y revelación, y se encuentran protegidos por la legislación sobre la propiedad intelectual. A menos que figure explícitamente en el contrato de licencia o esté permitido por la ley, no se podrá utilizar, copiar, reproducir, traducir, emitir, modificar, conceder licencias, transmitir, distribuir, exhibir, representar, publicar ni mostrar ninguna parte, de ninguna forma, por ningún medio. Queda prohibida la ingeniería inversa, desensamblaje o descompilación de este software, excepto en la medida en que sean necesarios para conseguir interoperabilidad según lo especificado por la legislación aplicable.

La información contenida en este documento puede someterse a modificaciones sin previo aviso y no se garantiza que se encuentre exenta de errores. Si detecta algún error, le agradeceremos que nos lo comunique por escrito.

Si este software o la documentación relacionada se entrega al Gobierno de EE.UU. o a cualquier entidad que adquiera las licencias en nombre del Gobierno de EE.UU. entonces aplicará la siguiente disposición:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Este software o hardware se ha desarrollado para uso general en diversas aplicaciones de gestión de la información. No se ha diseñado ni está destinado para utilizarse en aplicaciones de riesgo inherente, incluidas las aplicaciones que pueden causar daños personales. Si utiliza este software o hardware en aplicaciones de riesgo, usted será responsable de tomar todas las medidas apropiadas de prevención de fallos, copia de seguridad, redundancia o de cualquier otro tipo para garantizar la seguridad en el uso de este software o hardware. Oracle Corporation y sus filiales declinan toda responsabilidad derivada de los daños causados por el uso de este software o hardware en aplicaciones de riesgo.

Oracle y Java son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

Intel e Intel Xeon son marcas comerciales o marcas comerciales registradas de Intel Corporation. Todas las marcas comerciales de SPARC se utilizan con licencia y son marcas comerciales o marcas comerciales registradas de SPARC International, Inc. AMD, Opteron, el logotipo de AMD y el logotipo de AMD Opteron son marcas comerciales o marcas comerciales registradas de Advanced Micro Devices. UNIX es una marca comercial registrada de The Open Group.

Este software o hardware y la documentación pueden proporcionar acceso a, o información sobre contenidos, productos o servicios de terceros. Oracle Corporation o sus filiales no son responsables y por ende desconocen cualquier tipo de garantía sobre el contenido, los productos o los servicios de terceros a menos que se indique otra cosa en un acuerdo en vigor formalizado entre Ud. y Oracle. Oracle Corporation y sus filiales no serán responsables frente a cualesquiera pérdidas, costos o daños en los que se incurra como consecuencia de su acceso o su uso de contenidos, productos o servicios de terceros a menos que se indique otra cosa en un acuerdo en vigor formalizado entre Ud. y Oracle.

Tabla de contenidos

Prefacio	13
Destinatarios	13
Requisitos previos	13
Acerca de este manual	13
¿Cómo elijo una solución para satisfacer mis necesidades?	15
Accesibilidad a la documentación	15
Novedades	17
1. Introducción a la recuperación ante desastres	19
Definición del objetivo de tiempo de recuperación (RTO)	20
Definición del objetivo de punto de recuperación (RTO)	21
Manejo de interrupciones temporales	22
Concepto clave: recuperación de punto de sincronización	22
Relación de RPO con la recuperación de punto de sincronización	23
Planificación para la alta disponibilidad de datos (D-HA)	24
Cinta física de alta disponibilidad	25
Cinta virtual de alta disponibilidad	26
D-HA y recuperación de punto de sincronización	29
Realización de una verdadera recuperación ante desastres	29
Planificación de pruebas de DR	30
Movimiento de datos para pruebas de DR	31
Pruebas de DR con exportación/importación física	32
Pruebas de DR con CDRT	33
Pruebas de DR con replicación cruzada de sistemas Tape de VSM	35
2. Realización de exportaciones e importaciones físicas	37
Exportación e importación por clase de gestión	38
Ejemplo: exportación por clase de gestión a partir del sistema VSM de origen	38
Ejemplo: importación por clase de gestión en el sistema VSM de destino	39
Exportación e importación por clase de almacenamiento	41
Ejemplo: exportación por clase de almacenamiento a partir del sistema VSM de origen	41

Ejemplo: importación por clase de almacenamiento en el sistema VSM de destino	42
Exportación e importación para VLE	44
3. Uso de la función de almacenamiento externo de ELS	47
Preparación para almacenamiento externo de ELS	47
Creación de MVC para DR y LTR	49
Consideraciones sobre DELSCR al usar la función de almacenamiento	50
¿Qué ocurre cuando los volúmenes almacenados se devuelven al ACS?	50
Almacenamiento de MVC para DR	50
Almacenamiento básico de DR con MVC	51
Paso 1: Creación de VTV/MVC de almacén	51
Paso 2: Exportación de los MVC de almacén	51
Paso 3: (Opcional) Escritura de juegos de datos adicionales en la cinta de archivo de manifiesto	52
Paso 4: Expulsión de los MVC de almacén	53
Paso 5: Expulsión de volúmenes nativos (incluida la cinta del archivo de manifiesto)	53
Paso 6: Creación de una lista desplegable de volúmenes para devolución a partir del almacén	54
Paso 7: preparación de MVC almacenados para devolución	55
Paso 8: Preparación de volúmenes nativos almacenados para devolución	56
Paso 9: Introducción de volúmenes devueltos	57
Almacenamiento de DR de varias semanas con MVC	58
Almacenamiento de DR con MVC en una biblioteca remota	59
Almacenamiento de MVC para LTR	60
Expulsión de volúmenes específicos a un almacén local (planta)	61
4. Uso de replicación cruzada entre sistemas TapePlex en una solución de DR	63
Funcionamiento de CTR	63
Consideraciones de VTV de CRT de solo lectura	64
Configuración para CTR	65
Instalación: configuración e inicio de CTR	66
Definición de políticas para CTR	69
Políticas para el TapePlex emisor	69
Políticas para el TapePlex receptor	71
Uso de CTR cuando el sitio remoto no tiene LPAR	72

Uso de CTR como solución de DR	74
Uso de CTR para continuidad empresarial	74
Uso de CTR para reanudación de las actividades empresariales	76
Pruebas de recuperación ante desastres mediante la replicación cruzada entre sistemas TapePlex	77
Realización de pruebas de DR cuando el sitio de DR no tiene LPAR	78
Gestión de VTV replicados mediante replicación cruzada entre sistemas TapePlex (CTR)	79
5. Configuración de una biblioteca remota	81
Modificación del archivo SMC SCMDS	81
Actualización de la plataforma VTCS CONFIG para definir una biblioteca remota	82
Consideraciones sobre la agrupación de MVC	83
6. Uso de configuraciones de VTSS en cluster	85
¿Qué es un VTSS en cluster?	85
Requisitos de VTSS en cluster	86
Cómo funcionan las configuraciones de VTSS en cluster	88
Cómo funciona la conciliación de VTSS	90
Clusters unidireccionales y bidireccionales	91
Clusters unidireccionales	91
Cómo funcionan los clusters de VTSS unidireccionales	91
Clusters bidireccionales	92
Cómo funcionan los clusters de VTSS bidireccionales	92
Agrupación en clusters ampliada	94
Replicación sincrónica o asincrónica	95
Implementación de la replicación sincrónica	95
Implementación de replicación asincrónica con supervisión de trabajos	96
Agrupación en clusters con conexiones TCP/IP	97
El entorno de TCP/IP	97
Configuración de VTCS para CLINK TCP/IP	98
Sentencia CONFIG CLINK	98
7. Uso del software de prueba concurrente de recuperación ante desastres	99
Consideraciones sobre metadatos	100
¿Dónde obtiene la CDRT los datos del VTV?	101

¿Qué ocurre con los datos cuando finaliza la prueba?	101
Gestión de los recursos de CDRT	102
Recursos de volúmenes	102
Subagrupaciones reutilizables	102
Recursos de MVC	103
Recursos de VTSS	104
Recursos de VTSS no compartidos	104
VTSS en cluster en CDRT	104
Preparación de un cluster de VTSS para una prueba de DR:	105
Gestión del contenido de VTSS antes y después de la prueba de DR	105
Recursos de compartidos	106
Recursos de ACS	107
Restricciones de ACS sobre la prueba de DR	107
Requisitos del host de producción de DR	107
Recursos de VLE	108
Políticas de VTCS	108
Definición de MGMTCLAS/STORCLAS para VTSS no compartidos	109
Definición de MGMTCLAS para VTSS compartidos	109
Optimización del acceso a recursos de prueba y producción	109
Ejecución de una prueba de DR	110
Limpieza después de una prueba de DR	112
Limpieza después de una prueba de DR	112
Reanudación de las operaciones normales	113
Escenarios operativos	113
Escenario 1: Sitios de prueba y producción, ACS en cada sitio, VTSS de reserva en el sitio de prueba	114
JCL de muestra para el escenario 1	114
Escenario 2: Sitios de prueba y producción, ACS en cada sitio, toma de control del VTSS en el sitio de prueba	115
JCL de muestra para el escenario 2	116
Escenario 3: Sitios de prueba y producción, ACS en cada sitio, sin VTSS	117
JCL de muestra para el escenario 3	118
Escenario 4: VTSS en cluster con sitios de producción y prueba de DR	118
JCL de muestra para el escenario 4	119
Escenario 5: Sitios de prueba y producción, ACS y VLE en cada sitio	120
JCL de muestra para el escenario 5	121
Escenario 6: Sitios de prueba y producción, solo VLE en cada sitio	122
JCL de muestra para el escenario 6	123

Escenario 7: VTSS en cluster (sin cinta) con sitios de producción y prueba de DR	124
JCL de muestra para el escenario 7	124
8. Creación de puntos de recuperación del sistema en entornos de VSM	127
Ejemplos de puntos de control	128
Ejemplo 1: copias de MVC locales y copias de MVC remotas	128
Ejemplo 2: uso de CONFIG RECLAIM PROTECT	131
9. Uso de la VLE para recuperación ante desastres	133
Modo de producción normal	133
Ejecución de una prueba de DR con la VLE	134
Limpieza después de una prueba de DR con VLE	138
Uso de la VLE para continuidad empresarial	139
A. Ejemplos de VTSS en cluster	141
VTSS en cluster unidireccional	141
Configuración y gestión de un sistema de VTSS en cluster unidireccional	143
VTSS en cluster bidireccional	146
Configuración y gestión de un sistema de VTSS en cluster bidireccional	149
Agrupación en clusters ampliada	153
Configuración y gestión de un sistema de 3 VTSS en cluster	153
Cluster VSM5 a VSM5 con CLINK TCP/IP	157
Cluster VSM5 a VSM 6 con CLINK TCP/IP y VLE interconectadas	158
Cluster VSM 6 a VSM 6 “sin cinta” con CLINK TCP/IP	160
¿Debe usar una configuración unidireccional o bidireccional?	161
Glosario	163
Índice	177

Lista de tablas

6.1. Requisitos de VTSS en cluster	87
6.2. Requisitos de replicación sincrónica	88
6.3. Requisitos de la replicación mejorada	88
A.1. Valores de IPIF de CLINK para VSMPR1	157
A.2. Valores de IPIF de CLINK para VSMPR2	157
A.3. Valores de IPIF de CLINK para VSMPR1	159
A.4. Valores de IPIF de RTD para VSMPR1	159

Lista de ejemplos

4.1. Plataforma CONFIG para TapePlex A	67
4.2. Plataforma CONFIG para TapePlex B	68

Prefacio

StorageTek Enterprise Library Software (ELS) de Oracle es una solución compuesta por el siguiente software básico:

- Componente de gestión de almacenamiento (SMC) de StorageTek
- Componente de software de host (HSC) de StorageTek Software de control de cinta virtual (VTCS) de StorageTek
- Prueba concurrente de recuperación ante desastres (CDRT) de StorageTek

Además, se proporciona el siguiente software con el paquete de ELS:

- Library Content Manager (LCM) de StorageTek (antes, ExLM) LCM incluye una versión mejorada del producto que antes se conocía como "función de almacenamiento fuera del sitio".
- Componente de sistema de cliente para entornos de MVS (MVS/CSC) de StorageTek
- StorageTek LibraryStation

Destinatarios

Esta guía está destinada al personal del cliente o de StorageTek que es responsable de la planificación y la implementación de soluciones de recuperación ante desastres (DR) y gestión de datos fuera del sitio.

Requisitos previos

¶Para llevar a cabo las tareas que se describen en esta guía, debe comprender lo siguiente:

- Sistema operativo z/OS
- JES2 o JES3
- Enterprise Library Software (ELS)

Acerca de este manual

La *Guía de gestión de datos fuera del sitio y recuperación ante desastres de ELS* incluye lo siguiente:

- [Capítulo 1, *Introducción a la recuperación ante desastres*](#) , que ofrece una introducción a las soluciones de recuperación ante desastres y una descripción general de estas soluciones.
- Para obtener información acerca de la gestión de datos fuera del sitio, consulte:

- [Capítulo 2, Realización de exportaciones e importaciones físicas](#) , que ofrece toda la información necesaria acerca del uso de las funciones de EXPORTACIÓN e IMPORTACIÓN. La función de EXPORTACIÓN crea MVC portátiles en el sitio de origen que, luego, se expulsan del ACS de origen, se los transporta físicamente a otro sitio y se usa la función IMPORTACIÓN para llevarlos al sitio de destino.
- [Capítulo 3, Uso de la función de almacenamiento externo de ELS](#) describe el uso de esta función de DR que, a su vez, utiliza las funciones EXPORTACIÓN e IMPORTACIÓN, además de comandos y funciones adicionales para automatizar el proceso de almacenamiento de MVC fuera del sitio.
- [Capítulo 5, Configuración de una biblioteca remota](#) muestra un ejemplo de configuración con una biblioteca remota e información sobre cómo configurarla.
- [Capítulo 6, Uso de configuraciones de VTSS en cluster](#) proporciona todo lo que necesita saber acerca de la implementación de los VTSS en cluster, incluida la agrupación en clusters ampliada (tres VTSS más por cluster), además de las novedades de ELS 7.0 y superiores. Un VTSS en cluster no es, en sentido estricto, una solución de DR. No obstante, debido a las capacidades distribuidas, de redundancia y de copia de seguridad del VTSS en cluster, ofrece soluciones de DR. Por ejemplo, configuro clases de gestión y almacenamiento, de modo de obtener dos copias de MVC, una en el sitio local y otra en el remoto. Para obtener más información, consulte también [Apéndice A, Ejemplos de VTSS en cluster](#).
- Para obtener información sobre soluciones específicas de DR, consulte:
 - [Capítulo 4, Uso de replicación cruzada entre sistemas TapePlex en una solución de DR](#) ,, que describe cómo usar la función de replicación cruzada entre sistemas TapePlex para configurar una solución completa de recuperación ante desastres para los datos de cinta.
 - [Capítulo 7, Uso del software de prueba concurrente de recuperación ante desastres](#) ,, que indica cómo utilizar esta función para validar las soluciones de DR de su sitio. Tenga en cuenta que para ELS 7.2, CDRT admite Virtual Library Extension (VLE) 1.0 y superiores.
 - [Capítulo 8, Creación de puntos de recuperación del sistema en entornos de VSM](#) ,, que indica cómo implementar puntos de recuperación de DR en un entorno de VSM.
 - [Capítulo 9, Uso de la VLE para recuperación ante desastres](#) ,, que describe el uso de VLE (Virtual Library Extension) como solución de recuperación ante desastres que proporciona un método simplificado y sin interrupciones de realización de pruebas de DR.

En forma conjunta, en todos estos capítulos se analizan métodos de traslado de datos de una ubicación a otra y métodos de gestión de datos. Estos procesos se pueden utilizar con otros para DR, continuidad empresarial o reanudación de las actividades empresariales. La función de almacén de ELS le permite gestionar volúmenes de retención a largo plazo (LTR) y DR fuera de sitio, además de volúmenes manuales de bastidores (“planta”).

Por lo tanto, cuenta con una variedad completa de soluciones de gestión de datos fuera de sitio y DR. ¿Cómo elige la mejor opción para su sitio entre estas soluciones?

¿Cómo elijo una solución para satisfacer mis necesidades?

Las soluciones descritas en la *Guía de gestión de datos fuera del sitio y recuperación ante desastres de ELS* varían en cuanto al costo y los beneficios que ofrecen: la regla general es que las soluciones de mayor costo permiten recuperaciones más rápidas o ciclos de gestión de datos más breves. Por ejemplo, la exportación o la importación de MVC (opcionalmente, con la función de almacén externo de ELS) tiene un costo relativamente bajo, pero la recuperación lleva tiempo, simplemente, debido a la necesidad de mover los MVC de un sitio a otro.

Sin embargo, la replicación cruzada entre sistemas TapePlex (CTR) es relativamente costosa. Necesita dos sitios, varios VTSS, RTD y MVC en ambos sitios, etc. La CTR utiliza la agrupación en clusters del VTSS para copiar datos de enlaces de cluster FICON o ESCON directamente de un VTSS de un TapePlex a un VTSS de otro TapePlex. El movimiento de los datos (y los metadatos) se realiza de forma electrónica entre los sitios; por lo tanto, la copia de seguridad y la recuperación se realizan potencialmente mucho más rápido que con el almacén externo.

Accesibilidad a la documentación

Para obtener información sobre el compromiso de Oracle con la accesibilidad, visite el sitio web del Programa de Accesibilidad de Oracle en <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Acceso a My Oracle Support

Los clientes de Oracle que hayan contratado servicios de soporte electrónico pueden acceder a ellos mediante My Oracle Support. Para obtener información, visite <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> o, si tiene alguna discapacidad auditiva, visite <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>.

Novedades

En esta revisión, se incluyen las siguientes actualizaciones:

- Se agregó una nota a la exportación y la importación de MVC por clase de gestión.

Consulte " [Exportación e importación por clase de gestión.](#)"

- Se agregó una nota a la exportación y la importación de MVC por clase de almacenamiento.

Consulte " [Exportación e importación por clase de almacenamiento.](#)"

- Se agregó un procedimiento para importación y exportación para VLE.

Consulte " [Exportación e importación para VLE.](#)"

- Se agregó una nota a la descripción de VTSS en clusters que describe la relación de la replicación sincrónica y mejorada con el manejador de interrupciones faltante (MIH).

Consulte " [¿Qué es un VTSS en cluster?](#)"

- Se agregó una tabla de requisitos de replicación mejorada.

Consulte " [Requisitos de VTSS en cluster.](#)"

Capítulo 1. Introducción a la recuperación ante desastres

Las mejores prácticas de recuperación ante desastres (DR) empresarial consisten, principalmente, en el diseño y la implementación de sistemas de software y hardware con tolerancia a fallos que pueden sobrevivir a un desastre (“continuidad empresarial”) y reanudar las operaciones normales (“reanudación empresarial”) con una intervención mínima y, en condiciones ideales, sin pérdida de datos. Construir entornos con tolerancia a fallos para cumplir con los objetivos de DR empresarial y las limitaciones presupuestarias reales puede resultar costoso y demorar mucho tiempo; además, exige un fuerte compromiso de la empresa.

Los planes de DR suelen abordar uno o más de los siguientes tipos de desastres:

- Daños significativos de las instalaciones de TI debido a un desastre natural (terremoto, huracán, inundación, etc.) u otras causas (incendio, vandalismo, hurto, etc.).
- Pérdida significativa de servicios críticos de las instalaciones de TI, por ejemplo, pérdida de alimentación, refrigeración y acceso a la red.
- Pérdida de personal clave.

El proceso de planificación de DR comienza con la identificación y la caracterización de los tipos de desastres a los que debe sobrevivir una empresa para, luego, reanudar sus operaciones. El proceso de planificación identifica requisitos de alto nivel de continuidad empresarial (BC) y reanudación empresarial (BR), entre ellos, el nivel necesario de tolerancia a fallos. A partir de la planificación de DR se genera una arquitectura de recuperación y reanudación para sistemas, aplicaciones y datos con tolerancia a fallos, a fin de satisfacer estas necesidades sujetas a limitaciones establecidas. Normalmente, las limitaciones de DR son el objetivo de tiempo de recuperación (RTO), el objetivo de punto de recuperación y el presupuesto disponible. La arquitectura de DR junto con las limitaciones empresariales favorecen la creación de procedimientos de DR que integran todos los elementos del sistema de una manera verdaderamente integral para garantizar resultados predecibles en relación con el proceso completo de DR.

Los sistemas con tolerancia a fallos, por lo general, alcanzan la solidez y la capacidad de recuperación mediante la *redundancia*. Un sistema completamente redundante, que suele lograrse a un alto costo, no tiene un solo punto de fallo dentro de su arquitectura y puede

funcionar durante las operaciones y reanudarlas a partir del peor desastre posible dentro de sus límites. Los sistemas de control de vuelo de aviones y transbordadores espaciales son buenos ejemplos de sistemas completamente redundantes. Las aplicaciones de TI menos críticas, por lo general, utilizan sistemas menos resistentes y menos redundantes. La construcción de estos sistemas es menos costosa, pero necesariamente implicará una interrupción del servicio después del desastre. Durante esta interrupción, la empresa trabajará para restablecer sus datos, aplicaciones y sistemas recuperables.

En última instancia, la naturaleza de una empresa, las necesidades de los clientes y el presupuesto disponible para DR son los factores clave para la determinación de los requisitos de DR. Una solución integral de DR puede ser muy costosa, pero es necesario diseñarla. No puede tirar el dinero, el hardware ni el software ante un posible desastre y tener la esperanza de sobrevivir y reanudar sus operaciones empresariales. No obstante, aunque realice una planificación y un diseño inteligentes, es posible que se vea afectado por interrupciones más prolongadas del servicio, un servicio degradado, o ambos, hasta que se puedan reanudar todos los servicios, pero, de todas maneras, puede contar con una solución limitada y confiable de DR.

De todas maneras, debe entender que quizás *no* exista ningún nivel de planificación que permita anticipar **todos** los posibles escenarios de DR o responder a ellos. Por ejemplo, lo que comienza como un problema aparentemente trivial de un sistema se puede propagar, con el transcurso del tiempo, y afectar a otros sistemas de distintas maneras y, finalmente, causar un desastre para el cual no hay una posible recuperación. De manera similar, la capacidad de una empresa para cumplir con los acuerdos de servicio puede verse afectada si las suposiciones clave no resultan ciertas, por ejemplo, si el servicio o las piezas clave no se encuentran disponibles, o si la capacidad de prestación del servicio del proveedor de DR no es tan sólida como se promociona. No obstante, lo más importante que se debe tener en cuenta es que si ocurre un desastre que **supera** el peor escenario para el cual usted se preparó, quizás la recuperación no sea posible.

Definición del objetivo de tiempo de recuperación (RTO)

RTO es un objetivo de nivel de servicio del tiempo que lleva lograr la capacidad operativa deseada después de que ocurre un desastre. Por ejemplo, las necesidades de la empresa pueden determinar un RTO en el que todos los sistemas de producción funcionen al 80 % de la capacidad previa al desastre en el plazo de los 30 minutos posteriores a una interrupción no planificada del servicio de más de una hora (si no existiera capacidad de DC). El tiempo de procesamiento de RPO, la disponibilidad de personal calificado de TI y la complejidad de los procesos manuales de TI necesarios después de un desastre son ejemplos de limitaciones que pueden determinar el RTO. El RTO no se aplica a los sistemas con tolerancia completa a fallos, porque estos sistemas se recuperan implícitamente durante un desastre y después de él, sin interrupción del servicio.

Los planificadores de DR pueden establecer distintos RTO para algunos o todos los requisitos de BC definidos. Los diversos tipos de operaciones empresariales pueden exigir distintos

RTO, por ejemplo, RTO diferentes para sistemas en línea que para ventanas de lotes. Además, se pueden aplicar distintos RTO en las diversas etapas de un plan de DR en fases, en el cual cada fase tiene un RTO definido. Incluso una aplicación recuperable puede tener distintos RTO para cada uno de sus diversos niveles de servicio.

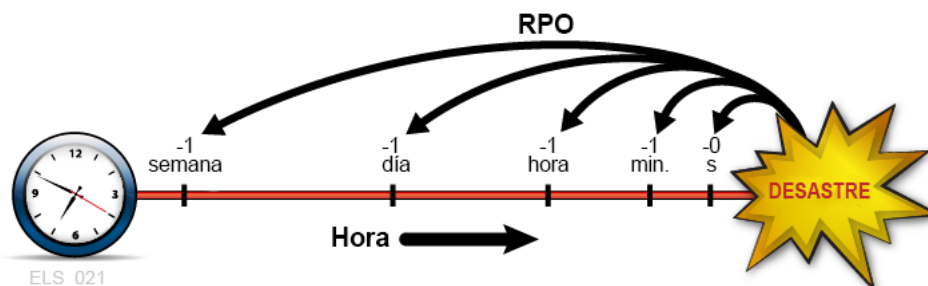
Los requisitos de disponibilidad de los datos de BC son sumamente importantes para la planificación de RTO. Cuando los datos que se deben introducir en el proceso de DR no están presentes en el sitio de recuperación ante desastres, el tiempo que lleve recuperar los datos en el sitio demorará el RTO. Por ejemplo, llevará tiempo recuperar los datos que residen en almacenes fuera del sitio. La recuperación puede continuar rápidamente si los datos de entrada actualizados están duplicados en el sitio de recuperación antes del comienzo de las operaciones de recuperación ante desastres.

Definición del objetivo de punto de recuperación (RTO)

El RPO es un objetivo de continuidad empresarial que indica el estado o el nivel de actividad de la empresa, y que se logra después de que el proceso de recuperación ante desastres ha restablecido todos los sistemas recuperables. Conceptualmente, el RPO es un estado de sincronización objetivo o una "reversión" que se conoce antes de que ocurra un desastre. Es decir, que el RPO es el punto de recuperación posterior a un desastre a partir del cual las aplicaciones recuperables interrumpidas pueden reanudar el procesamiento. Todas las transacciones que ocurren durante el intervalo entre el RPO y el momento del desastre son irre recuperables. El RPO no se aplica a sistemas con tolerancia completa ante fallos porque el desastre no afecta la continuidad empresarial de estos sistemas.

En [Figura 1.1, "Objetivos de punto de recuperación"](#), se ilustran los conceptos de RPO. En la imagen, se sugieren diversos puntos de recuperación para que tengan en cuenta los planificadores de DR. La planificación debe garantizar que el RPO deseado sea factible frente al RTO elegido y viceversa. Por lo general, los planes de recuperación ante desastres que exigen los RPO más próximos al momento de un desastre requieren una mayor tolerancia ante fallos y son más costosos de implementar, en comparación con los RPO más lejanos. Al igual que con los RTO, los planificadores de DR pueden establecer distintos RPO para diferentes requisitos de BC, fases de los planes de DR o niveles de servicio de las aplicaciones.

Figura 1.1. Objetivos de punto de recuperación



De manera más amplia, la planificación de RPO debe identificar todos los elementos complementarios que deben estar presentes para restablecer cada sistema recuperable, incluidos los datos, los metadatos, las aplicaciones, las plataformas, las instalaciones y el personal. La planificación también debe garantizar que estos elementos estén disponibles en el nivel deseado de actividad de la empresa para recuperación. Los requisitos de actividad de los datos de BC son especialmente cruciales para la planificación de RPO. Por ejemplo, si según los requisitos de BO, se requiere un RPO de una hora, cualquier dato o metadato que alimente el proceso de recuperación debe estar actualizado hasta el RPO, de lo contrario, no se puede alcanzar el RPO. Los procesos de DR de la organización especificarán los procedimientos necesarios para alcanzar todos los RPO definidos en el plazo de los RTO establecidos.

Los metadatos del sistema necesarios para la recuperación del RPO incluyen estructuras de catálogo de OS e información del sistema de gestión de cintas. Estos elementos de deben actualizar durante el proceso de recuperación ante desastres para activar todos los RPO elegidos. Por ejemplo, para garantizar la coherencia entre las diversas entradas de metadatos en el proceso de DR, los juegos de datos existentes que se recrearán en el RPO se deben descatalogar; los juegos de datos actualizados entre el RPO y el momento del desastre se deben restaurar a la versión que existía en el RPO o antes de este; y cualquier cambio de catálogo relacionado con cintas debe estar sincronizado con el sistema de gestión de cintas.

Manejo de interrupciones temporales

La recuperación ante desastres ofrece reparación para interrupciones muy prolongadas que pueden dejar a un sitio de producción inutilizable durante un período prolongado. Si bien el resto de esta introducción trata las prácticas de recuperación ante desastres, sería igualmente importante desarrollar procedimientos para mitigar las interrupciones relativamente breves que podrían afectar negativamente la producción si no se controlan. Considere, por ejemplo, una interrupción del servicio en que ciertas funciones de hardware o de red permanecen no disponibles durante una o dos horas, pero la producción puede continuar durante esta interrupción en “modo degradado” con unos pocos y rápidos ajustes temporales. Un procedimiento de interrupción temporal debe documentar cómo aislar el problema, qué cambios realizar, a quién notificar y cómo regresar al entorno operativo normal después del restablecimiento del servicio.

Concepto clave: recuperación de punto de sincronización

El reinicio de las aplicaciones de producción en los RPO definidos es una actividad clave que se realiza durante una verdadera recuperación ante desastres y durante las pruebas de DR. Los entornos de DR con mayor capacidad de recuperación garantizan que cada aplicación recuperable, ya sea de terceros o desarrollada internamente, aplique un requisito clave de DR, a saber: que la aplicación esté diseñada para reiniciarse de un momento planificado denominado "punto de sincronización", a fin de mitigar los efectos de una interrupción no programada durante su ejecución. Cuando una aplicación interrumpida se reinicia en un punto de sincronización, los resultados son los mismos que si la aplicación no hubiera sido interrumpida.

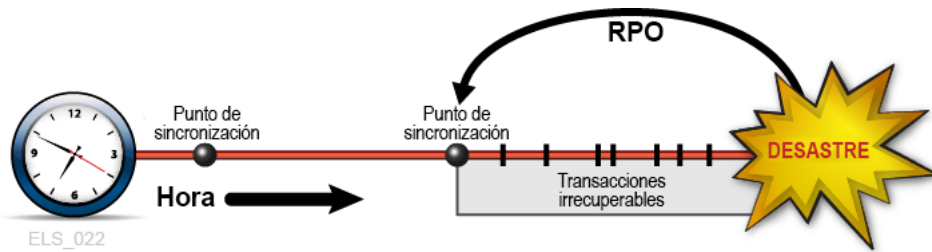
El procedimiento de reinicio para una aplicación recuperable depende de la naturaleza de la aplicación y sus entradas. A menudo, el procedimiento de reinicio de una aplicación para una verdadera recuperación ante desastres o pruebas de DR es el mismo procedimiento de que utiliza para reiniciar la aplicación en caso de que falle durante una ejecución de producción normal. En los casos en que es posible, la reutilización de los procedimientos de reinicio de producción para una verdadera recuperación ante desastres o pruebas de DR simplifica la creación y el mantenimiento de los procedimientos de DR, y aprovecha estos procedimientos probados. En el caso más sencillo, una aplicación recuperable constituye un paso de trabajo único con un solo punto de sincronización, que es el comienzo del programa invocado por ese paso. En ese caso, el procedimiento de recuperación puede ser tan sencillo como volver a ejecutar el trabajo interrumpido. Un procedimiento de inicio levemente más complejo podría implicar que la aplicación deba descatalogar todos los juegos de datos de salida durante su última ejecución y que, luego, deba reiniciar la aplicación.

Los procedimientos de reinicio para aplicaciones que tienen varios puntos de sincronización interna entre los cuales se puede elegir pueden no ser tan sencillos. Las aplicaciones que usan técnicas de punto de control/reinicio para implementar estos puntos de sincronización registran periódicamente su progreso y pueden, por ejemplo, usar la información registrada del punto de control para reiniciarse en el último punto de sincronización interna registrado antes de una interrupción. Los procedimientos de reinicio cumplirán con los requisitos de cada punto de sincronización. Mientras los puntos de control están en uso, los juegos de datos asociados con un punto de control no deben caducar, no se deben descatalogar ni se deben reutilizar mientras el punto de control sigue siendo válido para la recuperación de aplicaciones. Una manera sencilla de establecer un punto de sincronización para un paso de trabajo que modifica sus juegos de datos existentes es hacer una copia de seguridad de cada juego de datos modificable antes de ejecutar el paso. Estos juegos de datos de entrada modificables se pueden identificar fácilmente al buscar el atributo de JCL DISP=MOD en sentencias DD o en solicitudes de asignación dinámica. Si un paso de trabajo falla o se interrumpe, simplemente, deseche los juegos de datos de entrada modificados, restaure esos juegos de datos de entrada de las copias de seguridad y reinicie el paso a partir de las copias restauradas. Estas copias de seguridad también son útiles para reiniciar un paso de trabajo que falló o se interrumpió, el cual había caducado, descatalogado o reutilizado los originales.

Relación de RPO con la recuperación de punto de sincronización

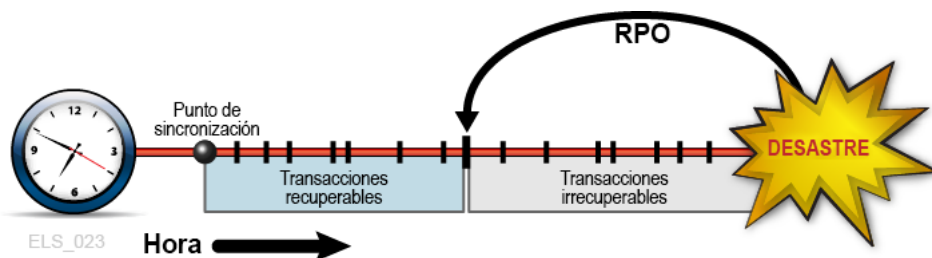
Cuando el RPO se alinea con un punto de sincronización, al realizar el procedimiento de reinicio de la aplicación que se desarrolló para este punto de sincronización, se reanudará la aplicación a partir de este origen como si no se hubiera producido una interrupción (Figura 1.2, “El RPO en el punto de sincronización”). Se supone que todas las transacciones procesadas después de este RPO hasta el desastre son irre recuperables.

Figura 1.2. El RPO en el punto de sincronización



En otros momentos, los requisitos de BC pueden justificar la ubicación del RPO entre los puntos de sincronización. En estos casos, la recuperación de punto de intersincronización se basa en datos complementarios que describen cualquier cambio o evento crítico relativo al estado de la aplicación que ocurre después del establecimiento del punto de sincronización más reciente. Considere, por ejemplo, el RPO de un minuto antes de un desastre. Suponga que se diseña una aplicación recuperable para usar puntos de control para registrar su progreso, pero suponga que la sobrecarga que implica el uso de estos puntos de control a intervalos de un minuto no se puede tolerar. Una solución sería tomar estos puntos de control con menor frecuencia y registrar todas las transacciones confirmadas entre los puntos de control. El log de esta transacción se convierte, luego, en datos de entrada complementarios utilizados por el proceso de recuperación del punto de control que se reiniciarán a partir de un RPO más allá del punto de sincronización más reciente. En este ejemplo, el procedimiento de reinicio de la aplicación accede a los datos más recientes del punto de control y aplica el log de la transacción complementaria para restablecer todas las transacciones confirmadas procesadas después del punto de control y antes del RPO (Figura 1.3, “RPO entre puntos de sincronización”). De esta manera, la recuperación de punto de sincronización puede alcanzar un RPO objetivo mediante los datos de entrada de varios orígenes. Se supone que todas las transacciones procesadas después del RPO hasta el desastre son irre recuperables.

Figura 1.3. RPO entre puntos de sincronización



Planificación para la alta disponibilidad de datos (D-HA)

Los datos suelen ser uno de los activos más valiosos de una empresa. La mayoría de las compañías cuidan rigurosamente los datos empresariales críticos y realizan inversiones adicionales para protegerlos contra la pérdida y para garantizar que los datos estén disponibles para su objetivo previsto cuando sea necesario. Una empresa que no puede

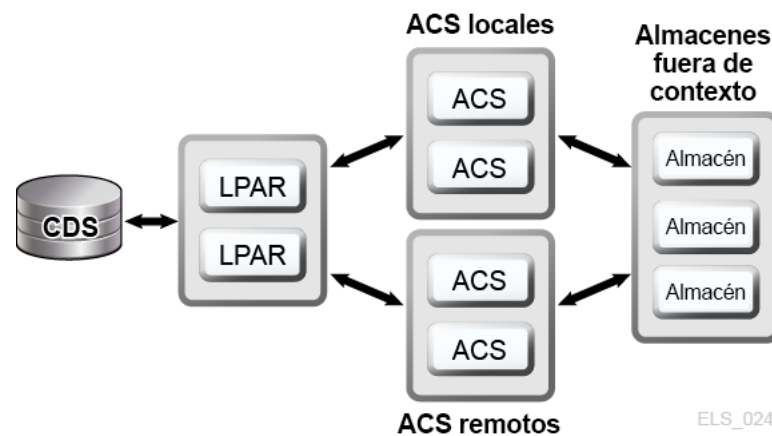
hacer frente a la pérdida de datos críticos puede sufrir consecuencias desastrosas. Quizás la manera más habitual de proteger los datos contra la pérdida es almacenar copias de datos críticos en distintos subsistemas o medios de almacenamiento, y almacenar algunas de estas copias en distintas ubicaciones físicas. Las copias almacenadas en medios de almacenamiento extraíbles, entre ellas, las cintas de cartuchos magnéticos, los CD-ROM y los DVD suelen almacenarse en ubicaciones de almacenamiento externas. Las copias adicionales también suelen almacenarse en el sitio, en instalaciones de TI en que las aplicaciones pueden procesar los datos. La creación y el almacenamiento de copias de datos críticos aumentan la redundancia de los datos y mejora la tolerancia a fallos. Para los medios extraíbles y, en particular, para las cintas de cartuchos magnéticos, el solo hecho de aumentar la redundancia de los datos no suele ser suficiente para garantizar que los datos también tengan alta disponibilidad para las aplicaciones que los utilizarán. Por ejemplo, el sistema VSM de Oracle para cintas virtuales de mainframe almacena datos en volúmenes de cintas virtuales denominados MVC. El sistema VSM puede hacer copias automáticamente para mejorar la redundancia de los datos y reducir el riesgo debido a un fallo de medios físicos o a un cartucho de cintas mal colocado. Un sistema de producción VSM utiliza varios componentes de hardware especializados para recuperar datos almacenados en un MVC, incluido un dispositivo de buffer de VTSS, una biblioteca de cintas automatizada y unidades de cinta conectadas a bibliotecas (denominadas RTD), que también se conectan al dispositivo de buffer de VTSS. Las aplicaciones host dependen de todos estos componentes de VSM que funcionan juntos para recuperar datos de los MVC. Si bien la mayoría de las personas no consideraría el simple fallo de un componente como un desastre equiparable a la pérdida de un centro de datos entero en un terremoto, ciertamente, podría ser imposible recuperar datos de un MVC si un único componente crítico de VSM falla sin una copia de seguridad, independientemente de cuantas copias redundantes de MVC existan. Por lo tanto, si bien la creación de copias de MVC es una de las mejores prácticas comprobadas para mitigar la vulnerabilidad y los riesgos, no siempre garantiza de forma suficiente la alta disponibilidad de los datos (D-HA) en la presencia de fallos. Los requisitos de D-HA son requisitos clave de continuidad empresarial para la planificación de DR. Por lo general, la D-HA se logra aumentando las redundancias para eliminar los puntos únicos de fallo que no permiten a las aplicaciones acceder a los datos durante los fallos del sistema. Por ejemplo, un sistema VSM que incluye componentes redundantes mejora la tolerancia a fallos del sistema VSM. La instalación de varios dispositivos VTSS, handbots redundantes de SL8500 y varias RTD tiene el propósito de eliminar los puntos únicos de fallo de VSM de la ruta de datos de la aplicación a los datos críticos almacenados en un MVC. La arquitectura de VSM está diseñada para admitir la agregación de componentes redundantes para aumentar la tolerancia a fallos promover la D-HA.

Cinta física de alta disponibilidad

Las soluciones de automatización de cintas de mainframe de Oracle activan la D-HA para aplicaciones de cinta físicas mediante el almacenamiento de copias redundantes de datos en distintos ACS dentro de un sistema TapePlex, es decir, dentro de un complejo de cintas asignado por un solo CDS. Por ejemplo, las aplicaciones que se ejecutan en instalaciones de TI con un solo sistema TapePlex pueden almacenar fácilmente copias duplicadas de juegos

de datos de cintas en uno o más ACS dentro de ese TapePlex. Esta técnica mejora la D-HA mediante la agregación de medios redundantes, transportes de cinta y bibliotecas de cintas automatizadas. En un caso simple, una aplicación almacena copias redundantes de juegos de datos críticos en dos cintas de cartuchos distintas en una sola biblioteca SL8500 con electrónica redundante, handbots duales en cada guía y dos o más transportes de cinta conectados a la biblioteca en cada guía, que son compatibles con los medios del juego de datos. Para eliminar la biblioteca SL8500 como un punto potencial único de fallo, se agrega una segunda SL8500 al ACS para almacenar copias aun más redundantes del juego de datos críticos. Para eliminar las instalaciones de TI como punto único de fallo, las copias de juegos de datos redundantes se pueden almacenar fuera del sitio o crear en un ACS remoto con transportes de cintas mediante una extensión de canal (Figura 1.4, “Configuración de cinta física FD-HA”).

Figura 1.4. Configuración de cinta física FD-HA

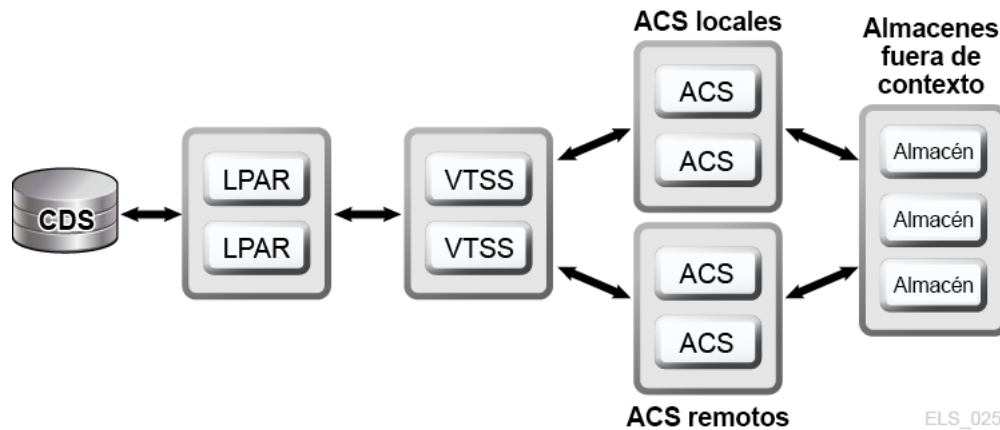


También puede hacer dos o más copias de cintas físicas en distintas ubicaciones físicas cuando una ubicación tiene su propio CDS independiente, es decir, cuando el hardware de cada ubicación representa un TapePlex independiente. Al usar la función de cliente/servidor del SMC y definir políticas para dirigir copias de juegos de datos a un TapePlex remoto, los trabajos pueden crear copias de cintas en un ACS de otro TapePlex sin cambios en JCL.

Cinta virtual de alta disponibilidad

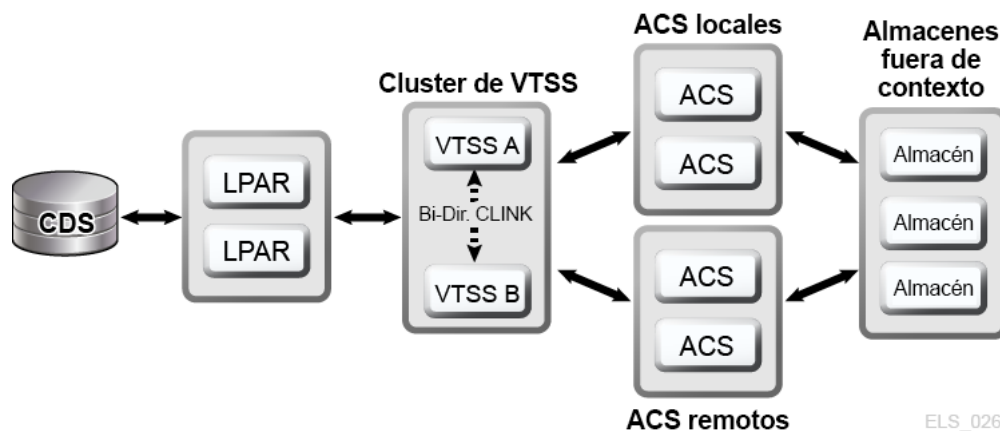
VSM ofrece tecnologías de agrupación en clusters y creación de varios complejos para replicación en MVC para activar la D-HA para cinta virtual de mainframe. La creación de varios complejos para replicación en VSM implica la creación de varias copias de MVC (por ejemplo, duplicados, cuadruplicados) en uno o más ACS para mayor redundancia (Figura 1.5, “Configuración de creación de varios complejos para replicación en VSM para D-HA”). Los ACS que reciben copias de las que se crean varios complejos para replicación pueden ser bibliotecas locales o ACS remotos con transportes de cinta mediante una extensión de canal. Las políticas de migración de VSM controlan el movimiento de VTV que residen en el buffer de VTSS a MVC locales o remotos, que pueden ser trasladados a almacenes externos.

Figura 1.5. Configuración de creación de varios complejos para replicación en VSM para D-HA



Un cluster de VSM comprende dos o más dispositivos VTSS (nodos) conectados en red para intercambio de datos en un enlace de comunicaciones (CLINK). Los CLINK son canales unidireccionales o bidireccionales. La configuración de cluster de VSM más sencilla consiste en dos nodos VTSS en el mismo TapePlex enlazado con un CLINK unidireccional, pero, habitualmente, se implementan CLINK bidireccionales (Figura 1.6, “Configuración de cluster de VSM de D-HA”). Cada nodo de cluster se puede ubicar en un sitio distinto. Las políticas de almacenamiento unidireccional de VSM controlan la replicación automática de volúmenes de cintas virtuales (VTV) de VTSS A a VTSS B en un CLINK unidireccional. Las políticas de almacenamiento bidireccional y los CLINK bidireccionales permiten que VTSS A se replique en VTSS B y viceversa.

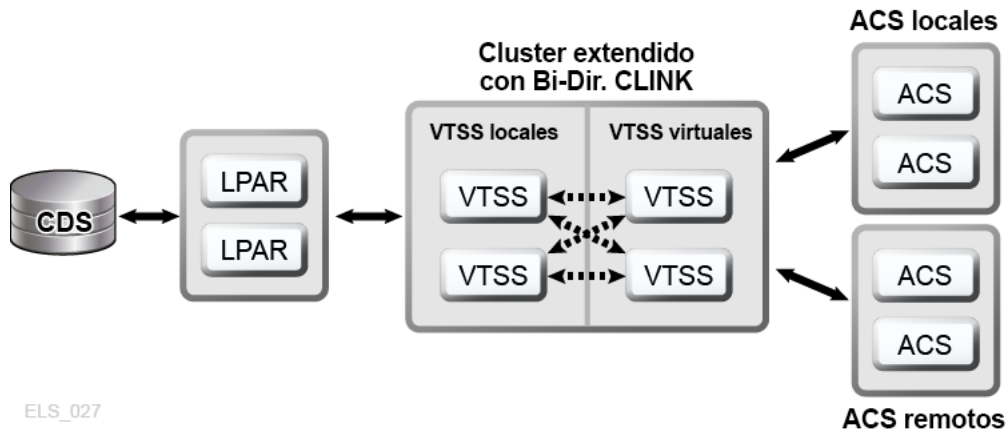
Figura 1.6. Configuración de cluster de VSM de D-HA



La agrupación en clusters ampliada de VSM permite la conectividad de varios a varios entre tres o más dispositivos VTSS en un TapePlex para alcanzar niveles de disponibilidad de datos aun más altos (Figura 1.7, “Configuración de cluster ampliado de D-HA (no se muestran los

almacenes externos)”). Como se muestra, la instalación de dispositivos de cluster de VTSS en uno o más sitios dentro de un TapePlex aumenta la redundancia, ya que elimina cada sitio como punto único de fallo.

Figura 1.7. Configuración de cluster ampliado de D-HA (no se muestran los almacenes externos)

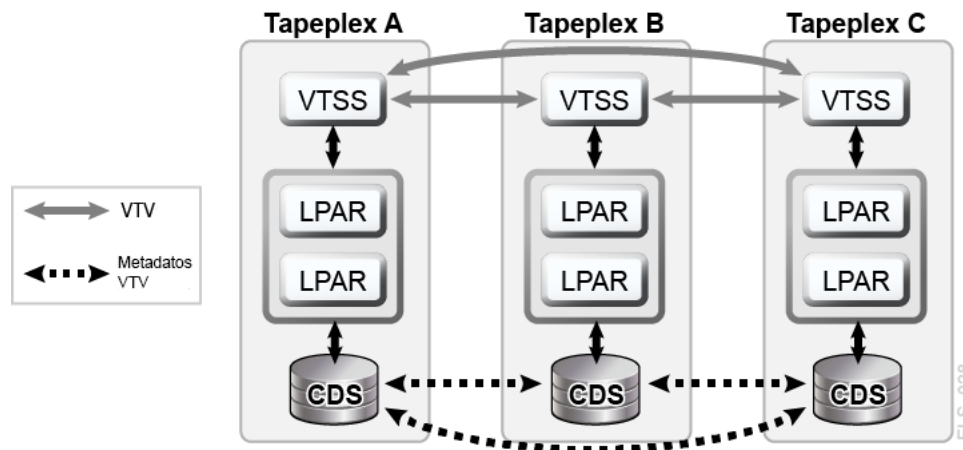


ELS_027

La LCM de Oracle simplifica los procesos de almacenamiento en sitios externos para volúmenes de MVC mediante la gestión del proceso de reciclaje entre los almacenes y las bibliotecas de producción. La función de almacenamiento de LCM programa la devolución de volúmenes de MVC almacenados cuando la cantidad de datos caducados supera un umbral especificado.

Un cluster de replicación cruzada entre sistemas TapePlex (cluster CTR) de VSM permite que los dispositivos de cluster de VTSS residan en distintos TapePlex y ofrece la capacidad de replicar VTV de un TapePlex a uno o más TapePlex distintos, lo que permite modelos de replicación de clusters de varios a varios en CLINK unidireccionales o bidireccionales (Figura 1.8, “Configuración de replicación cruzada entre sistemas TapePlex de VSM de D-HA”). El envío y la recepción de TapePlex puede estar ubicado en distintos sitios. Los VTV replicados se introducen en el CDS del TapePlex receptor como volúmenes de solo lectura. Esto ofrece una protección de datos segura contra la alteración que pueden causar las aplicaciones que se ejecutan en el TapePlex receptor. El CDS del TapePlex receptor también indica que las copias de VTV replicadas por CTR son propiedad del TapePlex emisor y, como protección adicional, la CTR garantiza que un TapePlex no pueda modificar ningún VTV que no le pertenezca.

Figura 1.8. Configuración de replicación cruzada entre sistemas TapePlex de VSM de D-HA



D-HA y recuperación de punto de sincronización

La creación de varias copias de volúmenes físicos (MVC o distintos de MVC) mejora la redundancia de los datos; no obstante, estas copias requieren consideraciones especiales para la recuperación de punto de sincronización. Lo más importante de la recuperación de punto de sincronización es garantizar que los datos creados en un punto de sincronización se mantengan en un estado de solo lectura mientras siguen siendo válidos para uso en recuperación ante desastres. Esto significa que las copias de volúmenes de cinta físicos que se pueden usar para recuperación ante desastres se deben mantener en el estado de solo lectura. Una manera de hacer esto es enviar estas copias a una ubicación de almacén externo donde no hay capacidad de procesamiento de cintas. Es importante tener en cuenta que las copias no protegidas que sufren alteraciones no se pueden utilizar para recuperación de punto de sincronización, porque el contenido actualizado deja de reflejar el punto de sincronización asociado. Los entornos de cintas virtuales agregan una dimensión adicional a la gestión de copias de varios volúmenes para recuperación de punto de sincronización. Pueden existir copias de VTV en varios buffers de VSM y en varios MVC al mismo tiempo. Incluso cuando todos los MVC de un determinado VTV se almacenan en un sitio externo, las copias de VTV que permanecen en el sitio en buffers de VSM pueden modificarse. No es posible usar una copia actualizada de un VTV que reside en un buffer para recuperación de punto de sincronización, a menos que este VTV pertenezca a un nuevo punto de sincronización que invalide las copias externas almacenadas para uso durante la recuperación ante desastres.

Realización de una verdadera recuperación ante desastres

El éxito de una operación de verdadera recuperación ante desastres yace en contar con un sitio de DR adecuado, personal capacitado, un procedimiento de DR comprobado, una carga de trabajo de producción recuperable con puntos de sincronización para cumplir con los RPO definidos y todos los datos de entrada y metadatos del sistema necesarios para cumplir con estos RPO. Los datos de entrada y los metadatos del sistema deben estar accesibles en el sitio de DR cuando se los necesita y deben estar disponibles en los niveles de actividad

necesarios. Con una planificación minuciosa, una preparación rigurosa y una ejecución de comprobada eficacia, las operaciones de verdadera recuperación ante desastres pueden fluir sin inconvenientes según lo planificado para lograr los RPO y los RTO definidos. Los datos de producción generados en el sitio de DR deben estar protegidos de manera adecuada mientras el sitio de DR funciona como sitio de producción. Suponga, por ejemplo, que la arquitectura de D-HA requiere una carga de trabajo de producción para replicar copias de datos redundantes en tres sitios remotos, y suponga que el sitio de DR es uno de estos sitios de replicación remotos antes de un desastre. Cuando el sitio de producción experimenta un desastre y su carga de trabajo se traslada al sitio de DR, este sitio ya no puede funcionar como sitio de replicación remoto para la carga de trabajo de producción que ahora se ejecuta localmente en dicho sitio. Para cumplir con el requisito de D-HA de tres sitios de replicación remotos, se debe agregar un nuevo tercer sitio de replicación remoto durante el período en que la producción se mantenga en el sitio de DR. Este ejemplo ilustra de qué manera un análisis exhaustivo de los requisitos de D-HA permite a los planificadores de DR abordar todos los requisitos críticos de D-HA que se deben cumplir cuando la producción se traslada a un sitio de DR. Un plan integral de DR comprende no solamente las actividades para restablecer la producción en el sitio DR, sino que también incluye el proceso de desocupación del sitio de DR mientras el sitio de producción se encuentra en reparación y hasta que quede listo para volver a usarse, si consideramos al sitio de DR solamente como un sustituto temporal para producción. Por ejemplo, cuando el sitio de producción está listo para reanudar las operaciones, los datos de producción se deben restablecer en dicho sitio. Los métodos incluyen la agrupación en clusters bidireccional entre el sitio de DR y el sitio de producción, lo que permite tiempo suficiente para que el trabajo de producción que se ejecuta en el sitio de DR vuelva a rellenar el sitio de producción anterior mediante la replicación de datos. No obstante, puede ser necesario, u oportuno o eficaz, simplemente transportar los MVC físicos nuevamente al sitio de producción restablecido. Los métodos elegidos dependerán de las necesidades de recuperación posteriores al desastre.

Planificación de pruebas de DR

La preparación para una verdadera recuperación ante desastres se evalúa probando la eficacia y la eficiencia de los procedimientos y sistema de DR para recuperar una carga de trabajo de producción en un sitio de prueba de DR designado. El entorno de prueba de DR debe ser un entorno de prueba de DR dedicado, pero, normalmente, es más económico compartir recursos entre los sistemas de prueba de DR y de producción. Las pruebas de DR que se realizan simultáneamente con la producción y que comparten el uso de recursos con la producción se conocen como "pruebas concurrentes de DR". Si una aplicación debe ejecutarse en paralelo en sistemas prueba de DR y de producción, los planificadores de DR deben garantizar que estas dos instancias de la aplicación no interfieran entre sí mientras se ejecutan de manera simultánea. El aislamiento de los sistemas de prueba de DR y de producción en LPAR independientes y la limitación del acceso a los datos de producción desde el sistema de prueba de DR suele proporcionar una separación suficiente. Las pruebas de DR se suelen realizar en etapas para permitir la realización de pruebas específicas de distintas aplicaciones en distintos momentos, en lugar de realizarse una prueba de recuperación de todo el entorno de producción de una vez. Las pruebas específicas son clave para reducir la cantidad de

recursos de hardware dedicado que se necesita para el sistema de prueba de DR. Por ejemplo, si las pruebas de DR de una aplicación recuperable requieren solo un pequeño subconjunto de recursos de VSM, esos recursos se pueden compartir entre los sistemas de prueba de DR y de producción, y se pueden reasignar al sistema de prueba de DR para el ciclo de prueba de DR. Este enfoque reduce el costo de hardware para el sistema de DR a riesgo de afectar el rendimiento del sistema de producción mientras se ejecuta la prueba de DR. No obstante, por lo general, un ciclo de prueba de DR dedica solo un pequeño porcentaje de recursos compartidos a la prueba de DR, y el entorno de producción disminuido no se ve afectado en gran medida por las pruebas simultáneas de DR. Sin embargo, algunas organizaciones tienen políticas que impiden afectar o alterar la producción para facilitar las pruebas de DR. Es posible que los auditores exijan una coincidencia exacta entre los resultados de pruebas de DR y de producción para certificar el proceso de DR. Una manera de cumplir con este requisito consta en establecer un punto de sincronización justo delante de una ejecución programada de producción, guardar una copia de los resultados de producción, recuperar la ejecución de producción en este punto de sincronización en el sitio de prueba de DR y comparar el resultado con los resultados de producción guardados. Cualquier diferencia entre los resultados destacará una brecha que se deberá investigar. La imposibilidad de detectar estas brechas oportunamente podría en riesgo poner la capacidad de recuperación ante desastres de una organización. Independientemente de si una prueba de DR está diseñada para recuperar una carga de trabajo compleja o una aplicación única, el proceso de prueba de DR se debe realizar con los mismos procedimientos que se utilizarían para una verdadera recuperación ante desastres. Esta es la única manera segura de comprobar la realización correcta de la prueba de DR.

Movimiento de datos para pruebas de DR

Hay dos métodos para organizar los datos de aplicación para pruebas de DR en un sitio de prueba de DR: movimiento físico de datos y movimiento electrónico de datos. El movimiento físico de datos implica el transporte de cartuchos de cinta físicos al sitio de prueba de DR en un proceso que se describe a continuación y que se conoce como "importación/exportación física". El movimiento electrónico de datos utiliza unidades de cinta remotas, RTD remotas y técnicas de cluster de VSM para crear copias de los datos de aplicación en un sitio de prueba de DR. Ambos métodos de movimiento de datos permiten la realización de pruebas de DR; no obstante, el movimiento electrónico de datos evita la transferencia física de los datos y los posibles problemas de la pérdida de cintas, entre otros. La transferencia electrónica también mejora el tiempo de acceso a los datos, ya que los coloca donde se los necesita para una verdadera recuperación ante desastres o los almacena provisionalmente en un buffer de VSM antes de un ciclo de prueba de DR. El movimiento electrónico de datos para volúmenes virtuales se puede realizar dentro de un solo TapePlex mediante el uso de una agrupación en clusters ampliada de VSM, o entre dos TapePlex mediante la replicación cruzada entre sistemas TapePlex. Para los datos que se encuentran dentro de un solo TapePlex, el software de prueba concurrente de recuperación ante desastres (CDRT) de Oracle simplifica las pruebas de DR.

Pruebas de DR con exportación/importación física

Suponga que desea realizar pruebas de DR para una aplicación de producción que utiliza cinta virtual y física. Su objetivo es probar esta aplicación en el sitio de prueba de DR repitiendo una ejecución de producción reciente y verificando que el resultado de la prueba coincida con el resultado de la producción reciente. Como parte de la preparación, deberá guardar copias de los juegos de datos de entrada utilizados por la ejecución de producción y una copia del resultado de la producción para comparación. Suponga que el sitio de prueba de DR está aislado y no comparte equipos con la producción. Usted podría realizar la prueba de DR mediante este proceso de exportación/importación física.

Sitio de producción:

1. Haga una copia de los VTV y los volúmenes físicos requeridos.
2. Exporte esas copias de VTV.
3. Expulse las copias asociadas de MVC y las copias de volúmenes físicos del ACS de producción.
4. Transporte los MVC y los volúmenes físicos expulsados al sitio de prueba de DR.

Sitio de prueba de DR:

1. Introduzca los volúmenes transportados en el ACS de DR.
2. Sincronice los catálogos de OS y el sistema de gestión de cintas con los volúmenes introducidos.
3. Importe los datos de VTV/MVC.
4. Ejecute la aplicación.
5. Compare los resultados.
6. Expulse todos los volúmenes introducidos para esta prueba.
7. Transporte los volúmenes expulsados nuevamente al sitio de producción.

Sitio de producción:

1. Introduzca los volúmenes transportados nuevamente en el ACS de producción.

Este proceso permite que las pruebas de DR continúen de forma segura y en paralelo con la producción, ya que el sistema de prueba de DR está aislado del sistema de producción. El sistema de prueba de DR tiene su propio CDS, y el proceso de prueba de DR, como se indica anteriormente, introduce información de los volúmenes en el CDS de prueba de DR a modo de preparación para la prueba de DR. Esto permite que la aplicación recuperada se pruebe con los mismos nombres de juegos de datos y volúmenes que utiliza en producción. Para los juegos de datos de cintas virtuales, la función de almacenamiento del software LCM de Oracle simplifica la colocación de los VTV en los MVC y agiliza los pasos circulares que requieren exportar y expulsar volúmenes en el sitio de producción, importar esos volúmenes en el sitio de prueba de DR y expulsar dichos volúmenes para moverlos nuevamente al sitio de producción. La exportación/importación física le genera gastos al sitio por la manipulación

de cintas físicas y gastos de mensajería por el transporte de cartuchos de cinta entre los sitios de producción y de prueba de DR. Los datos confidenciales que se transportan mediante servicio de mensajería se deben transportar en cartuchos de cinta cifrados. Los plazos de las pruebas de DR se ven afectados por el tiempo que se dedica a transportar y manipular los cartuchos de cinta entre los sitios.

Pruebas de DR con CDRT

Si se planifica y se cuenta con recursos suficientes de hardware en los sitios de producción y de DR, la CDRT combinada con el movimiento electrónico de datos puede eliminar la necesidad de transportar cartuchos de cinta física al sitio de DR y permite la realización concurrente de pruebas de DR de una manera más económica que si se mantiene un sitio de prueba de DR dedicado y aislado. La CDRT permite la realización de pruebas de DR de prácticamente cualquier carga de trabajo, configuración, RPO o RTO imaginable. El proceso de prueba de DR incluirá unos pocos pasos adicionales para iniciar la CDRT y una limpieza después de la prueba de DR. Antes de ejecutar una prueba de DR con CDRT, debe mover electrónicamente todos los datos de aplicación y metadatos del sistema (información de catálogo de OS e información del sistema de gestión de cintas) necesarios para la prueba al sitio de prueba de DR. Puede mover datos de aplicación electrónicamente mediante la agrupación en clusters de VSM o la migración de copias de VTV a los MVC del sitio de prueba. A continuación, puede usar la CDRT para crear un CDS especial para el sistema de prueba de DR que refleje el CDS de producción. Los sistemas de producción y de prueba de DR son entornos separados, y el entorno de prueba de DR utilizará el CDS especial de prueba de DR en lugar del CDS de producción. Debido a que la CDRT crea el CDS de prueba de DR a partir de información del CDS de producción, contiene metadatos de todos los volúmenes que se movieron electrónicamente al sitio de prueba de DR antes de la prueba de DR. Esto permite a las aplicaciones de DR utilizar los mismos números de serie de volúmenes y nombres de juegos de datos utilizados en la producción. La CDRT aplica restricciones operativas sobre el sistema de prueba de DR para evitar que el entorno de DR interfiera en el entorno de producción. Puede reforzar estas protecciones mediante las capacidades VOLPARM/POOLPARM de ELS para definir rangos de volser separados para los MVC y reutilizar los VTV para uso exclusivo de CDRT. La CDRT permite al sistema de prueba de DR leer los MVC de producción y escribir en su propia agrupación dedicada de MVC, que se borra lógicamente después de cada prueba de DR. Para aplicaciones de cinta virtual, la CDRT requiere al menos un dispositivo VTSS dedicado durante el ciclo de prueba de DR. Estos VTSS dedicados se pueden reasignar temporalmente desde la producción para facilitar una prueba de DR, y la prueba de DR y el sistema VSM pueden acceder a los ACS de producción en paralelo con la carga de trabajo de producción. En [Figura 1.9, “Cluster de producción con un VTSS2 de nodo de cluster remoto en el sitio de prueba de DR”](#) y [Figura 1.10, “Configuración de producción con un VTSS2 prestado para pruebas de DR de CDRT”](#), se muestra la división de un cluster de producción de VSM para prestar un dispositivo de cluster al sistema de prueba de DR de la CDRT, en este caso, VTSS2, en el sitio de prueba de DR. Cuando este cluster se divide, debe modificar las políticas de producción para sustituir la migración por la replicación, de modo que VTSS1 cree copias redundantes de VTV en el sitio de DR en ACS01, y de modo que VTSS1 no se llene hasta el límite de su capacidad

mientras el cluster se divide. VTSS2 se deja fuera de línea para la producción y se coloca en línea para la LPAR de prueba de DR. En [Figura 1.9, “Cluster de producción con un VTSS2 de nodo de cluster remoto en el sitio de prueba de DR”](#), CDRT ha creado el CDS de prueba de DR a partir de una copia remota del CDS de producción. Solo el sistema de producción puede acceder a los volúmenes de VTSS1 y ACS00 durante el ciclo de prueba de DR, y solo el sistema de prueba de DR puede acceder a VTSS2. Los sistemas de producción y de prueba de DR comparten el acceso concurrente a volúmenes de ACS01. En [Figura 1.9, “Cluster de producción con un VTSS2 de nodo de cluster remoto en el sitio de prueba de DR”](#) y [Figura 1.10, “Configuración de producción con un VTSS2 prestado para pruebas de DR de CDRT”](#), se mantiene una copia remota del CDS de producción en el sitio de prueba de DR, por ejemplo, mediante reflejo remoto, para garantizar que un CDS de producción actualizado esté disponible en el sitio de DR para el uso de una verdadera recuperación ante desastres. No obstante, se debe tener en cuenta que el CDS de prueba de DR creado por la CDRT a partir de la copia remota de CDS es una versión especial de la prueba de DR del CDS de producción que solo puede usar la CDRT. Antes de volver a conformar el cluster de producción, después de la finalización del ciclo de prueba de DR, el VTSS de DR se debe purgar para evitar la pérdida de datos de producción, como sucedería si VTSS2 tuviera una versión más reciente de un VTV que también existiera en VTSS1. También debe modificar las políticas de producción para realizar una reversión de una migración a una replicación cuando se vuelve a conformar el cluster. Si no es posible dividir un cluster de producción como se muestra aquí, otra posibilidad es mantener un VTSS independiente en el sitio de DR exclusivamente para la realización de pruebas de DR. En este caso, los VTV necesarios para la prueba para se recuperarán a partir de las copias de MVC.

Figura 1.9. Cluster de producción con un VTSS2 de nodo de cluster remoto en el sitio de prueba de DR

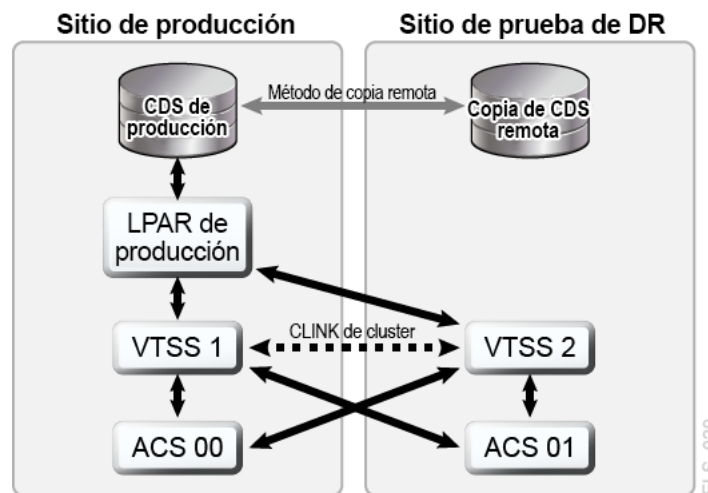
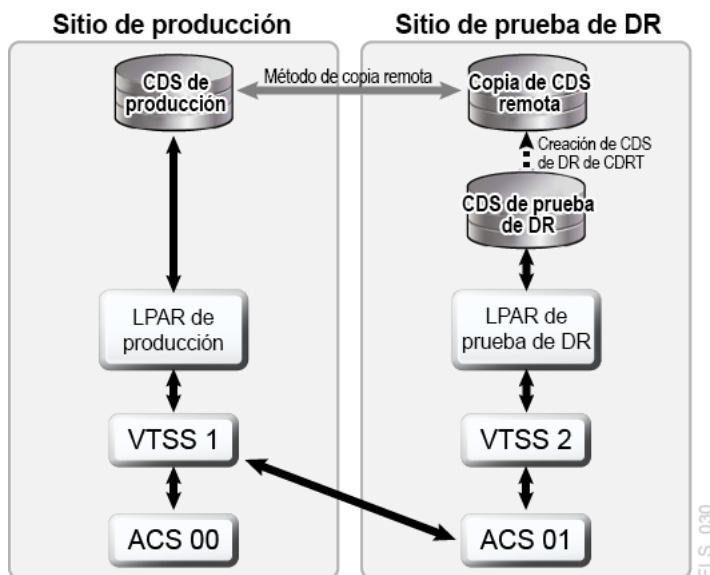
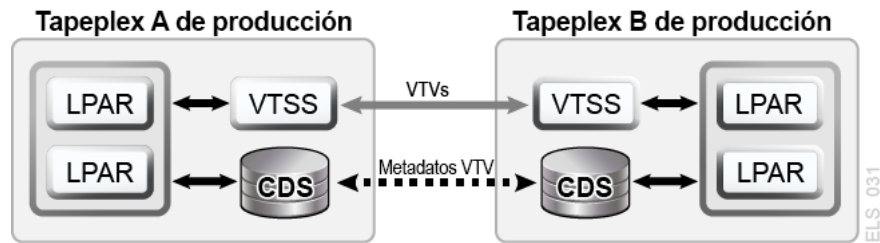


Figura 1.10. Configuración de producción con un VTSS2 prestado para pruebas de DR de CDRT



Pruebas de DR con replicación cruzada de sistemas Tape de VSM

La replicación cruzada entre sistemas TapePlex de VSM permite diseños de TapePlex de producción simétricos y en cluster que facilitan la realización de pruebas de DR sin el uso de CDRT, sin la necesidad de hardware de VTSS dedicado exclusivamente a las pruebas de DR y sin alterar el entorno de producción para realiza las pruebas de DR. Por ejemplo, CTR permite a cada TapePlex de producción replicar datos en los otros TapePlex de producción en el mismo cluster de CTR. Los clusters de punto a punto de CTR de producción pueden eliminar la necesidad de un sitio de prueba de DR dedicado. CTR permite distintos tipo de diseños de TapePlex en cluster y facilita las pruebas de DR de cualquier configuración o carga de trabajo de producción, con cualquier RPO o RTO factible. En un ejemplo simple, un cluster bidireccional de CTR une dos sistemas TapePlex simétricamente y cada TapePlex replica datos en el otro TapePlex (Figura 1.11, “Cluster de CTR de producción simétrica para pruebas de DR”). Un TapePlex receptor introduce un VTV replicado en su CDS en el estado de solo lectura y marca el VTV para indicar que es de propiedad del TapePlex emisor. En este ejemplo, las pruebas de DR para una aplicación de TapePlex A implican la replicación de datos de aplicación en el TapePlex B y la recuperación de la aplicación en el TapePlex B.

Figura 1.11. Cluster de CTR de producción simétrica para pruebas de DR

La simetría del diseño de este cluster de punto a punto significa que la aplicación recuperada que se está probando en el sitio del mismo nivel se ejecuta de la misma manera durante una prueba de DR que durante la producción. El CDS del mismo nivel contiene toda la información replicada de volúmenes necesaria para las pruebas de DR, que continúan en paralelo con la producción, y el mismo hardware de VTSS admite el uso concurrente de las cargas de trabajo de prueba de DR y producción. Los clusters de producción de VTSS pueden existir dentro de cada TapePlex, y no es necesario dividirlos para compartir el uso del hardware en distintos sistemas TapePlex para pruebas de DR. El TapePlex de producción en el cual se realiza la prueba de DR de la aplicación no puede modificar ningún VTV replicado por CTR; por lo tanto, todos los datos de producción replicados se mantienen completamente protegidos durante el ciclo de prueba de DR. Más importante aún, las pruebas de DR basadas en CTR garantizan que un procedimiento de prueba de DR validado tendrá idénticos resultados durante una verdadera recuperación ante desastres. El software de host de SMC emitirá un mensaje si se realiza un intento de actualizar un VTV replicado por CTR, lo cual permite identificar la aplicación como aquella que modifica un juego de datos de entrada existente. Mediante el seguimiento de las mejores prácticas para la gestión de puntos de sincronización que se indican anteriormente, debe poder garantizar que el entorno de producción guarde una copia de este juego de datos antes de que la aplicación lo modifique, en caso de que se necesite una copia de seguridad para la recuperación de punto de sincronización.

Capítulo 2. Realización de exportaciones e importaciones físicas

Las funciones EXPORT e IMPORT le proporcionan las herramientas para crear MVC físicamente portátiles. En el sitio de origen, utiliza EXPORT para consolidar los VTV en los MVC (si es necesario) y genera un archivo de manifiesto que describe el contenido del MVC (VTV en el MVC). A continuación, expulsa los MVC del sitio de origen, los transporta físicamente al sitio de destino y, luego, mediante IMPORT, los importa con el archivo de manifiesto para actualizar el CDS con información en los MVC y VTV importados. Tenga en cuenta que puede importar los VTV en un CDS sin necesidad de que el VTCS esté activo. A continuación, introduce los MVC en el sitio de destino.

Nota:

- Si decide devolver los MVC exportados al sistema de origen, no se requiere ningún procesamiento especial del VTCS, simplemente, introduzca los MVC en un LSM en el sistema de origen.
- Por cada VTV importado, las únicas copias de MVC que se crearán serán copias de los MVC que hayan sido exportados e importados mediante las mismas sentencias. Esto tiene una importancia especial al importar VTV duplicados. Dicho VTV solo tendrá copias en ambos MVC después de la importación si ambos MVC están presentes en el mismo archivo de manifiesto y se importan como resultado de la misma sentencia IMPORT.

Puede realizar la exportación mediante alguno de los siguientes métodos generales:

- Exportación por VTV o clase de gestión, que consolida los VTV seleccionados en un nuevo conjunto de MVC. Dado que la consolidación lleva tiempo y necesita recursos de VTSS, la opción preferida consiste en realizar la exportación por MVC o clase de almacenamiento. Para obtener más información, consulte "[Exportación e importación por clase de gestión.](#)"
- Exportación por MVC o clase de almacenamiento. Una exportación por clase de almacenamiento o MVC no requiere el procesamiento posterior de la consolidación de los VTV, y no requiere movimiento de datos. La exportación, simplemente, crea un archivo de manifiesto que describe el contenido del MVC seleccionado. Para obtener más información, consulte "[Exportación e importación por clase de almacenamiento.](#)"

Nota:

Si exporta por:

- **Volsers de VTV:** use un informe de TMS, LCM o VTVRPT para identificar los VTV necesarios.
- **Volsers de MVC:** use un informe de LCM o MVCRPT para identificar los MVC necesarios.
- **Clase de gestión:** revise las definiciones de su clase de gestión para identificar las clases de gestión necesarias.
- **Clases de almacenamiento:** revise las definiciones de su clase de almacenamiento para identificar las clases de almacenamiento necesarias.

Exportación e importación por clase de gestión

En los siguientes ejemplos, se muestran exportaciones e importaciones de MVC por clase de gestión.

Nota:

Cuando MGMTclas o un VTV realizan una exportación, el CDS no mantiene la información de MVC en el registro del VTV. El MVC se marca como (E)xport (Exportado) en este caso.

Ejemplo: exportación por clase de gestión a partir del sistema VSM de origen

Esta es la fase de "envío" de la exportación/importación, donde se empaquetan los datos deseados y se mueven fuera del sistema VSM de origen.

Para realizar una exportación a partir de un sistema VSM de origen, haga lo siguiente:

1. Identifique la clase de gestión que se utiliza para exportación.
2. Exporte por clase de gestión:

```
//EXPORT EXEC PGM=SLUADMIN, PARM=' MIXED' REGION=6M
//STEPLIB DD DSN=h1q .SEALINK, DISP=SHR
//MOVE1 DD DSN=h1q.REMOTE2, DISP=(, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, (1, 1), RLSE),
// DCB=( RECFM=FB, LRECL=80, BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
EXPORT MGMT (PAY, ACCOUNT) MANIFEST(MOVE1)
```

En este ejemplo, el archivo de manifiesto de salida es MOVE1 y se necesita para la importación. Dado que exportó por clase de gestión, EXPORT consolida (hace copias de) los VTV seleccionados en los MVC de exportación. Los MVC de exportación se marcan como de solo lectura y se exportan al CDS; a partir de entonces, se encuentran disponibles para expulsión desde un sistema LSM de origen. Estas copias consolidadas

del VTV son copias adicionales y no se registran en el CDS. Por ejemplo, si el VTV se duplicó antes de la exportación, el CDS registra ambas copias duplicadas, pero la tercera copia adicional usada para consolidación no se registra en el CDS. Por lo tanto, los VTV originales siguen estando disponibles para el sistema de origen. Puede usar los datos de los VTV originales o almacenarlos temporalmente para, luego, reutilizarlos.

Atención:

Programa la exportación para un momento en que los datos exportados no se estén actualizando.

3. Elimine los MVC para exportación de la agrupación de MVC. Para obtener más información, consulte *Gestión de HSC y VTCS*.
4. Expulse los MVC para exportación del LSM de un sistema VSM de origen. Para obtener más información, consulte *Gestión de HSC y VTCS*.
5. Si lo desea, reutilice los VTV exportados en el sistema de origen, déjelos en estado no disponible, o reutilice los datos que contienen.

Después de la exportación, el sistema de origen conserva los registros del CDS de los VTV y los MVC exportados. Los MVC de exportación se marcan como exportados y como de solo lectura en el CDS del sistema de origen. En este momento, tiene dos opciones, según el motivo por el cual ha exportado los VTV:

- **Si exportó los VTV para proporcionar una copia de seguridad a un segundo sitio**, deje los VTV en modo de solo lectura en el CDS del sistema de origen, de modo que no se puedan actualizar.
- **Si moverá de forma definitiva los VTV exportados a un segundo sitio**, reutilícelos o déjelos en estado no disponible de alguna otra manera en el CDS del sistema de origen. Use las utilidades de reutilización del HSC o la función SYNCVTV del LCM para reutilizar los VTV exportados.

Ejemplo: importación por clase de gestión en el sistema VSM de destino

Ya ha pasado un mes y, finalmente, está listo para la parte de "recepción" (importación) de la operación de exportación/importación.

Para realizar una importación en un sistema VSM de destino, haga lo siguiente:

1. Si los VTV y los MVC que importa no están en el CDS del sistema de destino, vuelva a establecer las definiciones *POOLPARAM/VOLPARAM* para agregar estos volsers, como se describe en *Configuración de HSC y VTCS*.

Si es necesario, aumente el tamaño del CDS en el sistema VSM de destino. Para obtener más información, consulte *Configuración de HSC y VTCS* o *Gestión de HSC y VTCS*.

¿Qué ocurriría si hubiera volsers de VTV duplicados en los sistemas de origen y destino? En general, debe hacer lo siguiente:

- Si el sistema de origen tiene más VTV actuales con los mismos volsers que en el sistema de destino, especifique *REPLACE(ALL)*.
 - Si está moviendo los VTV del sistema de origen al de destino (primera exportación/importación), especifique *REPLACE(NONE)*. En este caso, debe decidir qué hacer con los VTV duplicados caso por caso.
2. Introduzca los MVC para importación en el LSM de un sistema VSM de destino.

Para obtener más información, consulte *Gestión de HSC y VTCS*. Se recomienda ubicar los MVC físicamente antes de usar *IMPORT* para indicarle al CDS que tiene nuevos MVC y VTV.

3. De manera opcional, realice una ejecución de "validación" de *IMPORT*:

```
//IMPORT EXEC PGM=SLUADMIN,PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
      IMPORT MANIFEST(MOVE1) NOUPDATE
```

En este ejemplo, se muestra un JCL que ejecuta la utilidad *IMPORT*, donde:

- El archivo de manifiesto es el manifiesto de exportación especificado en el paso 2.
- *REPLACE(NONE)* (el valor por defecto) especifica que el VTCS no sobrescribe los VTV duplicados.
- *IMMDRAIN(NO)* (el valor por defecto) especifica que el VTCS no drena todos los VTV importados al espacio del VTSS.
- *NOUPDATE* especifica que el CDS no se actualiza (ejecución de validación únicamente).
- *INACTCDS* no se especifica; por lo tanto, el HSC está activo.

La realización de una ejecución de validación es opcional, pero se recomienda, porque usted realmente desea saber qué ocurrirá antes de pulsar el botón. Revise minuciosamente el informe de importación. ¿Le gusta lo que ve? Continúe con el Paso 4.

Nota:

- *IMPORT* es válido únicamente si *FEATures VSM(ADVGMGT)* está especificado.
 - Asegúrese de que el CDS de "destino" cuente con las mismas características (activadas en función del nivel de CDS) que el CDS de "origen". Por ejemplo, si el CDS de "origen" tiene activados tamaños de páginas grandes de VTV y se han creado VTV de 2/4 Gb, el CDS de "destino" debe tener las mismas capacidades, de lo contrario, la importación falla.
-

4. Realice una ejecución real de *IMPORT*:

```
//IMPORT EXEC PGM=SLUADMIN,PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1,DISP=SHR
```



```
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(MOVE1) REPLACE(ALL)
```

En este ejemplo, se muestra un JCL que ejecuta la utilidad *IMPORT*, donde, al igual que en la ejecución de "validación", *REPLACE(NONE)* (el valor predeterminado) especifica que el VTCS no sobrescribe los VTV duplicados.

Nota:

¿Qué ocurre si desea devolver los MVC al sistema de origen? De ser así, puede especificar *IMMDRAIN(YES)* para drenar los MVC de importación.

5. Ajuste las definiciones del VTV según sea necesario. Por ejemplo, debe definir nuevos VTV en el TMS del sistema de destino.
6. Realice una de las siguientes acciones:
 - Opcionalmente, ejecute *MVCMaint* para que los MVC importados admitan escritura. El VTCS importa los MVC en modo de solo lectura. Para que admitan escritura, debe ejecutar *MVCMaint* y especificar *READONLY OFF*. Lo más probable es que desee usar los nuevos MVC en el sistema de destino, y este es el primer paso para poder hacerlo.

A continuación, agregue los MVC importados a la nueva agrupación de MVC, como se describe en *Gestión de HSC y VTCS*. En este momento, los MVC se pueden recuperar y drenar, y se pueden realizar migraciones hacia ellos.

- Si especificó *IMMDRAIN(YES)* en el paso 4, puede devolver los MVC al sistema de origen.

Exportación e importación por clase de almacenamiento

En los siguientes ejemplos, se muestran operaciones de exportación e importación por clase de almacenamiento a partir de un VSM de origen.

Nota:

Cuando *STORclas* o *MVC* realizan una exportación, el CDS mantiene la información de MVC en el registro del VTV. El MVC se marca como (R)ead-Only (Solo lectura) en este caso.

Ejemplo: exportación por clase de almacenamiento a partir del sistema VSM de origen

Esta es la fase de "envío" de la exportación/importación, donde se empaquetan los datos deseados y se mueven fuera del sistema VSM de origen.

Para realizar una exportación a partir de un sistema VSM de origen, haga lo siguiente:

1. Identifique la clase de almacenamiento que se utiliza para exportación.
2. Exporte por clase de almacenamiento:

```
//EXPORT EXEC PGM=SLUADMIN, PARM=' MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//MOVE2 DD DSN=h1q.REMOTE2, DISP=(, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, (1, 1), RLSE),
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
EXPORT STOR(OFF1, OFF2) MANIFEST(MOVE2)
```

En este ejemplo, el archivo de manifiesto de salida es *MOVE2* y se necesita para la importación. Dado que exportó por clase de almacenamiento, el sistema crea un archivo de manifiesto, pero no se produce la consolidación del VTV. Los MVC de exportación se marcan como de solo lectura en el CDS; a partir de entonces, se encuentran disponibles para expulsión desde un sistema LSM de origen. Los VTV que residían en los MVC eliminados del LSM se pueden seguir usando, siempre que residan en otros MVC.

Atención:

Programa la exportación para un momento en que los datos exportados no se estén actualizando.

3. Elimine los MVC para exportación de la agrupación de MVC. Para obtener más información, consulte *Gestión de HSC y VTCS*.
4. Expulse los MVC para exportación del LSM de un sistema VSM de origen. Para obtener más información, consulte *Gestión de HSC y VTCS*.
5. Si lo desea, reutilice los VTV exportados en el sistema de origen, déjelos en estado no disponible, o reutilice los datos que contienen.

Después de la exportación, el sistema de origen conserva los registros del CDS de los VTV y los MVC exportados. Los MVC de exportación se marcan como exportados y como de solo lectura en el CDS del sistema de origen. En este momento, tiene dos opciones, según el motivo por el cual ha exportado los VTV:

- **Si exportó los VTV para proporcionar una copia de seguridad a un segundo sitio**, deje los VTV en modo de solo lectura en el CDS del sistema de origen, de modo que no se puedan actualizar.
- **Si moverá de forma definitiva los VTV exportados a un segundo sitio**, reutilícelos o déjelos en estado no disponible de alguna otra manera en el CDS del sistema de origen. Use las utilidades de reutilización del HSC o la función SYNCVTV del LCM para reutilizar los VTV exportados.

Ejemplo: importación por clase de almacenamiento en el sistema VSM de destino

Ya ha pasado un mes y, finalmente, está listo para la parte de "recepción" (importación) de la operación de exportación/importación.

Para realizar una importación en un sistema VSM de destino, haga lo siguiente:

1. Si los VTV y los MVC que importa no están en el CDS del sistema de destino, vuelva a establecer las definiciones *POOLPARAM/VOLPARAM* para agregar estos volsers, como se describe en *Configuración de HSC y VTCS*.

Si es necesario, también aumente el tamaño del CDS en el sistema VSM de destino. Para obtener más información, consulte *Configuración de HSC y VTCS* o *Gestión de HSC y VTCS*.

¿Qué ocurriría si hubiera volsers de VTV duplicados en los sistemas de origen y destino? En general, debe hacer lo siguiente:

- Si el sistema de origen tiene más VTV actuales con los mismos volsers que en el sistema de destino, especifique *REPLACE(ALL)*.
 - Si está moviendo los VTV del sistema de origen al de destino (primera exportación/importación), especifique *REPLACE(NONE)*. En este caso, debe decidir qué hacer con los VTV duplicados caso por caso.
2. Introduzca los MVC para importación en el LSM de un sistema VSM de destino.

Para obtener más información, consulte *Gestión de HSC y VTCS*. ¿Puede ver lo que está ocurriendo aquí? Se recomienda ubicar los MVC físicamente antes de usar *IMPORT* para indicarle al CDS que tiene nuevos MVC y VTV.

3. De manera opcional, realice una ejecución de "validación" de *IMPORT*.

```
//IMPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
      IMPORT MANIFEST(REMOTE1) NOUPDATE
```

En este ejemplo, se muestra un JCL que ejecuta la utilidad *IMPORT*, donde:

- El archivo de manifiesto es el manifiesto de exportación especificado en el paso 2.
- *REPLACE(NONE)* (el valor por defecto) especifica que el VTCS no sobrescribe los VTV duplicados.
- *IMMDRAIN(NO)* (el valor por defecto) especifica que el VTCS no drena todos los VTV importados al espacio del VTSS.
- *NOUPDATE* especifica que el CDS no se actualiza (ejecución de validación únicamente).
- *INACTCDS* no se especifica; por lo tanto, el HSC está activo.

La realización de una ejecución de validación es opcional, pero se recomienda, porque usted realmente desea saber qué ocurrirá antes de pulsar el botón. Revise minuciosamente el informe de importación. ¿Le gusta lo que ve? Continúe con el Paso 4.

Nota:

- *IMPORT* es válido únicamente si *FEATures VSM(ADVGMGT)* está especificado.
- Asegúrese de que el CDS de "destino" cuente con las mismas características (activadas en función del nivel de CDS) que el CDS de "origen". Por ejemplo, si el CDS de "origen" tiene activados tamaños de páginas grandes de VTV y se han creado VTV de 2/4 Gb, el CDS de "destino" debe tener las mismas capacidades, de lo contrario, la importación falla.

4. Realice una ejecución real de *IMPORT*:

```
//IMPORT EXEC PGM=SLUADMIN, PARM=' MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
      IMPORT MANIFEST(REMOTE1)
```

En este ejemplo, se muestra un JCL que ejecuta la utilidad *IMPORT*, donde, al igual que en la ejecución de "validación", *REPLACE(NONE)* (el valor predeterminado) especifica que el VTCS no sobrescribe los VTV duplicados.

Nota:

¿Qué ocurre si desea devolver los MVC al sistema de origen? De ser así, puede especificar *IMMDRAIN(YES)* para drenar los MVC de importación.

-
5. Ajuste las definiciones del VTV según sea necesario.
 6. Realice una de las siguientes acciones:
 - Opcionalmente, ejecute *MVCMaint* para que los MVC importados admitan escritura. El VTCS importa los MVC en modo de solo lectura. Para que admitan escritura, debe ejecutar *MVCMaint* y especificar *READONLY OFF*. Lo más probable es que desee usar los nuevos MVC en el sistema de destino, y este es el primer paso para poder hacerlo.
- A continuación, agregue los MVC importados a la nueva agrupación de MVC, como se describe en *Gestión de HSC y VTCS*. En este momento, los MVC se pueden recuperar y drenar, y se pueden realizar migraciones hacia ellos.
- Si especificó *IMMDRAIN(YES)* en el paso 4, puede devolver los MVC al sistema de origen.

Exportación e importación para VLE

Puede exportar VTV y MVC a un VLE compartido entre dos sistemas Tapeplex siempre y cuando tengan agrupaciones de MVC separadas.

Para realizar la exportación a un VLE conectado a dos sistemas Tapeplex, use lo siguiente como ejemplo:

1. Supongamos que Tapeplex A tiene un valor de MVCPOOL de A00-A10, y que Tapeplex B tiene un valor de MVCPOOL de B00-B10.
2. Exporte A00-A10 a la agrupación de MVC Tapeplex A de VLE.
3. Importe la agrupación de MVC A00-A10 a Tapeplex B mediante el comando `IMPORT REPLACE(UPDATED)`.

Capítulo 3. Uso de la función de almacenamiento externo de ELS

La función de almacenamiento externo de ELS (almacén de ELS) reemplaza y mejora, en gran medida, a su predecesora, la función de almacenamiento fuera del sitio de VSM. La función de almacén de ELS proporciona las siguientes mejoras para volúmenes de cinta real para almacenamiento:

- Utiliza el CDS de HSC para almacenar datos del volumen almacenado y datos de almacén. El uso de la información de almacenamiento de CDS, en lugar del TMS elimina:
 - El riesgo de errores humanos al devolver volúmenes al entorno automatizado
 - Volúmenes atascados en la ubicación de almacén cuando faltan en la lista desplegable que se genera
 - El riesgo de dejar accidentalmente volúmenes de almacén devueltos en el entorno automatizado
- Utiliza LCM para gestionar el proceso de almacenamiento, que incluye tres métodos de almacenamiento:
 - Almacenamiento de MVC para recuperación ante desastres (DR). Para obtener más información, consulte "[Almacenamiento de MVC para DR.](#)"
 - Almacenamiento de MVC para retención a largo plazo (LTR). Para obtener más información, consulte "[Almacenamiento de MVC para LTR.](#)"
 - Expulsión de volúmenes a un almacén de planta. Para obtener más información, consulte "[Expulsión de volúmenes específicos a un almacén local \(planta\).](#)"

Preparación para almacenamiento externo de ELS

El primer paso consiste en definir el área de volumen almacenado del CDS del HSC. Para hacerlo, debe ejecutar la utilidad SLUADMIN SET VAULTVOL. Por ejemplo:

```
SET VAULTVOL NBRVOLS(40000)
```

Nota:

- Si necesita agregar más volúmenes de almacén después de la definición inicial, debe hacerlo mediante *MERGEcds*; por lo tanto, permita volúmenes suficientes en la definición inicial para cubrir sus necesidades por anticipado.
 - Debe contar con bloques FREE (LIBRES) suficientes en el CDS para alojar los volúmenes que planea almacenar (volúmenes de cinta real), además de espacio adicional para crecimiento. Consulte *Configuración de HSC y VTCS* para obtener información sobre cómo calcular el espacio para volúmenes de almacén y consulte *Gestión de HSC y VTCS* para obtener información sobre cómo ampliar el CDS si no es lo suficientemente grande para alojar los volúmenes almacenados.
-

El paso dos consiste en definir los almacenes que contienen los volúmenes almacenados. Para hacerlo, debe ejecutar la utilidad *SLUADMIN SET VAULT* para cada almacén. Por ejemplo:

```
SET VAULT ADD NAME(DRVLT1) SLOTS(10000) DESC('DR Vault')
SET VAULT ADD NAME(LTRVLT1) SLOTS(20000) DESC('LTR Vault')
SET VAULT ADD NAME(FLOOR) SLOTS(500) DESC('Floor Vault')
```

Nota:

La cantidad total de ranuras en todos los almacenes que define no puede superar la cantidad de volúmenes especificados en la sentencia *VAULTVOL*.

Mientras que HSC define los almacenes y los volúmenes almacenados, LCM los gestiona. Tenga en cuenta, especialmente, los siguientes parámetros de almacenamiento de LCM:

GRACEPERIOD

La cantidad de días que transcurren entre el momento en que se selecciona un volumen para devolución a partir del almacén y el momento en que realmente se devuelve al entorno automatizado. El período de gracia proporciona un margen de seguridad, de modo que lleguen nuevos volúmenes al almacén antes de que se devuelvan volúmenes antiguos. Si no se especifica, el valor predeterminado es tres días.

DEFAULT

DEFAULT se excluye mutuamente con *GRACEPERIOD* y, por lo general, se utiliza para el almacén de "planta" (bastidor manual) que contiene automáticamente todos los volúmenes expulsados del entorno automatizado mediante *LCM EJECT(ASNEEDED)*. Otros volúmenes que se expulsan del ACS también se pueden asignar a este almacén, por ejemplo, un volumen que está activo pero que ya no es un buen candidato para automatización. Generalmente, este es un almacén definido que, en realidad, está constituido por bastidores en la planta del centro de datos. *DEFAULT* es un período de gracia de cero días para permitir que estos volúmenes se vuelvan a introducir en el ACS en cualquier momento.

También se debe tener en cuenta que hay opciones de expulsión estándares disponibles para todos los volúmenes de almacén. Entre ellas, se incluyen los CAP que se utilizarán, la definición de un mensaje de expulsión, el modo de las expulsiones y si las expulsiones deben estar en orden de serie de volúmenes o ranuras.

Creación de MVC para DR y LTR

Siempre que los MVC se almacenan para DR, debe haber, como mínimo, dos copias de cada VTV para separar los MVC, uno de los cuales permanece en el sitio, mientras el otro MVC se expulsa y se coloca en un almacén externo. Esto se hace mediante la asignación de dos clases de almacenamiento en la clase de gestión que se asigna al VTV.

Por lo tanto, el desafío es que desea proteger los datos, pero desea usar el espacio del MVC de la manera más económica posible, como se indica a continuación:

- **Defina la mínima cantidad posible de clases de almacenamiento.** El exceso de clases de almacenamiento, por lo general, significa demasiados MVC y/o MVC con pocos VTV.
- **Use la mínima cantidad posible de VTSS para crear MVC.** De ser posible, use solo un VTSS para crear MVC de almacén.

¿Qué otras consideraciones hay para la creación y el almacenamiento de MVC? Considere lo siguiente:

- **En primer lugar, los VTV deben migrar a los MVC de almacén tan pronto como sea posible**, de modo que se puedan expulsar y mover al almacén, porque estos VTV, generalmente, no se usan como entrada para otros pasos de trabajo.
- **En segundo lugar, dado que los VTV de DR caducan en distintos momentos**, considere la posibilidad de agrupar los VTV con fechas de caducidad similares en los MVC. No obstante, para reducir la cantidad total de MVC que se crean para enviar al almacén, limite la cantidad de estos grupos. Dado que ya habrá actividad para consolidar los VTV en menos MVC, el beneficio es mínimo más allá de dos grupos, uno para los VTV con períodos de caducidad muy cortos, por ejemplo, en el plazo de los próximos siete días, y uno para todos los demás volúmenes. Los VTV con control de catálogo se deben considerar como parte de un segundo grupo, ya que no hay manera de saber la fecha de caducidad real.
- **En tercer lugar, si bien no es necesario tener almacenes separados para DR y LTR**, puede resultar ventajoso tener MVC de DR ubicados en un almacén independiente. Esto permite que estos volúmenes se agrupen y se envíen al sitio de DR cuando es necesario.

Para los datos de LTR, las consideraciones son levemente distintas. En primer lugar, por definición, los datos de LTR no caducarán por un período prolongado. Por lo tanto, a diferencia de los datos de DR, que caducan con el transcurso del tiempo, los MVC de LTR no se fragmentarán. Se realizará un procesamiento periódico inicial de estos MVC para lograr que los MVC almacenados se llenen tanto como sea posible, al igual que con los MVC de DR; no obstante, una vez que estén suficientemente llenos, estos MVC permanecerán estáticos. No hay necesidad de tener más de una sola clase de almacenamiento para estos volúmenes. No obstante, debe tener en cuenta que es posible que desee migrar inmediatamente algunos de datos de LTR, mientras que otros datos se pueden migrar a medida que la automigración de VTCS los seleccione. Por lo tanto, se recomienda tener una clase de almacenamiento pero dos clases de gestión para datos de LTR.

Consideraciones sobre DELSCR al usar la función de almacenamiento

El parámetro *DELSCR* de la sentencia *MGMTclas* se utiliza para especificar si VSM debe suprimir los VTV reutilizados, donde *DELSCR(YES)* hace que VSM suprima los VTV reutilizados, lo cual libera espacio del buffer del VTSS y espacio del MVC. Considere la posibilidad de especificar *DELSCR(YES)* para las clases de gestión de DR y LTR. Si especifica *DELSCR(YES)*, use solamente *SYNCVTV* de LCM para sincronización de volúmenes reutilizables. Para obtener más información acerca del uso de LCM para gestionar la sincronización de volúmenes reutilizables, consulte la *Guía del usuario de LCM*.

¿Qué ocurre cuando los volúmenes almacenados se devuelven al ACS?

El proceso de introducción de HSC se ha modificado para la función de almacenamiento de ELS, de modo que cada volumen que se introduce se verifique para determinar si el volumen es un volumen almacenado externamente. Para estos volúmenes almacenados, se realizarán una de dos acciones, en función del campo de fecha de devolución del registro de almacén de CDS:

- Si la fecha de devolución ya ha pasado, el volumen se introduce, los metadatos de los volúmenes almacenados a partir del proceso de expulsión se restauran, y el volumen se elimina de los registros de almacén.
- Si la fecha de devolución todavía **no** ha pasado o no se ha establecido una fecha de devolución para el volumen, el volumen se introduce, los metadatos de los volúmenes almacenados a partir del proceso de expulsión se restauran, pero el volumen se deja en los registros de almacén y se expulsa automáticamente durante el siguiente proceso de expulsión. ¿Por qué ocurre esto? Hay varias razones. Las dos razones más comunes son que el volumen se devolvió para algún tipo de proceso de recuperación de datos o que se extrajo el volumen incorrecto del almacén (lo cual ocurre con mucha frecuencia). Independientemente del motivo, el volumen pertenece al almacén y se devolverá a esta ubicación y se restablecerá su estado protegido.

Se debe tener en cuenta que este es el proceso para los volúmenes almacenados en un almacén físico. Para los volúmenes almacenados en una biblioteca remota, el proceso es levemente distinto; consulte " [Almacenamiento de DR con MVC en una biblioteca remota.](#)"

Almacenamiento de MVC para DR

En un escenario de DR, tiene los objetivos empresariales generales de optimizar el uso del buffer del VTSS, garantizar una migración rápida de datos críticos y, al mismo tiempo, mantener la disponibilidad de los datos.

Almacenamiento básico de DR con MVC

De acuerdo con este enfoque, se crean volúmenes de DR todos los días; por lo tanto, el procesamiento para moverlos al almacén de DR se ejecuta todos los días para garantizar que los MVC se muevan a un sitio externo de manera segura y se protejan.

Nota:

- Todos los VTV creados para fines de DR y todas las cintas nativas, incluida la cinta del archivo de manifiesto creada en "[Paso 2: Exportación de los MVC de almacén](#)", son controlados por el TMS del sitio (en cuanto a la caducidad de los volúmenes). Este proceso se limita al almacenamiento de los MVC relacionados y los volúmenes nativos seleccionados involucrados.
 - Los MVC solo se pueden asignar a un almacén. Para asignar un volumen a un nuevo almacén, el volumen primero se debe eliminar de una asignación previa a un almacén.
 - En "[Paso 7: preparación de MVC almacenados para devolución](#)", comienza el procesamiento periódico, que permite el reciclaje de los MVC de DR que se fragmentaron o se llenaron solo parcialmente en el momento de la creación. Esto se realiza mediante drenajes "lógicos" de los MVC almacenados que usan las copias de los VTV todavía actuales ubicados en los MVC locales. El procesamiento periódico minimiza la cantidad de volúmenes totales del almacén, al mismo tiempo que garantiza que toda la actividad relacionada se reduzca al mínimo, lo cual se logra mediante el uso de criterios de selección adecuados al entorno correspondiente. Después de un drenaje "lógico" correcto del MVC almacenado, se establece la fecha de devolución de ese volumen en el CDS. En el caso de los volúmenes nativos, como las cintas del archivo de manifiesto, los volúmenes que han pasado a tener el estado reutilizable en el TMS se seleccionan y se configuran para devolución. Es posible decidir con qué frecuencia ejecutar el procesamiento periódico (se puede ejecutar diariamente o mensualmente).
-

Paso 1: Creación de VTV/MVC de almacén

Los VTV de almacén de DR se crean mediante una clase de gestión que apunta a dos clases de almacenamiento: una que crea los MVC que quedarán dentro del entorno local y otra que crea los MVC que se almacenan. Por ejemplo:

```
STOR NAME(DRLOC)ACS(00) MEDIA(STK1RD)
STOR NAME(DRVLT1) ACS(00) MEDIA(STK1RD)
```

Paso 2: Exportación de los MVC de almacén

Los MVC de almacén se exportan mediante un archivo de parámetros LCM, como se muestra en el siguiente ejemplo.

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT1')
  NoSync
  GracePeriod(3)
;
Action
  Export
  Control(Serial )
```

```

MVC
DSN(DRVAULT.MANIFEST)
Storageclass(DRVLT1)
Vault('DRVLT')
;

```

En este ejemplo:

- La sentencia *OPTIONS* especifica *NOSYNC* y *NOTMS*, porque el almacenamiento no usa información de TMS para almacenamiento y no se necesitan metadatos de TMS.
- La sentencia *VAULT* especifica *DRVLT* como el almacén de DR.
- La sentencia *ACTION EXPORT* especifica:
 - Exportar MVC por volser.
 - Crear un archivo de manifiesto de exportación (*DRVAULT.MANIFEST*), en este caso, un volumen del ACS que se expulsa y se almacena con los MVC de DR.
 - Apuntar a la clase de almacenamiento de almacén creada en "[Paso 1: Creación de VTV/MVC de almacén.](#)"
 - Definir el almacén de DR (*DRVLT*) y asignarle MVC no asignados anteriormente al almacén.

Nota:

Los MVC exportados se marcan como de solo lectura durante el procesamiento de la exportación.

Puede crear varias clases de almacenamiento de almacén (por ejemplo, para separar los MVC con VTV con distintas fechas de caducidad). Si desea asignar distintas clases de almacenamiento de almacén al mismo almacén, puede hacerlo en una sola sentencia *ACTION EXPORT*. Por ejemplo, las siguientes sentencias asignan clases de almacenamiento *DRVLT1* y *DRVLT2* al mismo almacén (*DRVLT*):

```

Action
Export
Control( Serial )
MVC
DSN(DRVAULT.MANIFEST)
Storageclass(DRVLT1
              DRVLT2)
Vault('DRVLT')
;

```

Paso 3: (Opcional) Escritura de juegos de datos adicionales en la cinta de archivo de manifiesto

Después de crear la cinta de archivo de manifiesto en "[Paso 2: Exportación de los MVC de almacén](#)," puede ejecutar un trabajo que copie el juego de datos de control (CDS) del HSC, el catálogo de TMS, el catálogo del sistema y otros juegos de datos significativos de un momento determinado en la cinta del archivo de manifiesto para proporcionar puntos de DR adicionales.

Paso 4: Expulsión de los MVC de almacén

Los MVC de almacén se expulsan mediante un archivo de parámetros LCM, como se muestra en el siguiente ejemplo.

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Eject
  When(
    (inLsm)
    and
    (VaultName EQ 'DRVLT')
    Control(Serial)
    Ejmsg('Move to DR Vault')
  )
;
```

En este ejemplo:

- La sentencia *OPTIONS* especifica *NOSYNC* y *NOTMS*, porque el almacenamiento no usa información de TMS para almacenamiento y no se necesitan metadatos de TMS.
- La sentencia *VAULT* especifica *DRVLT* como el almacén de DR.
- La sentencia *ACTION EJECT* especifica:
 - Expulsar los MVC asignados a *DRVLT*; expulsar por volser.
 - El mensaje de expulsión.

Paso 5: Expulsión de volúmenes nativos (incluida la cinta del archivo de manifiesto)

Los volúmenes nativos (incluida la cinta del archivo de manifiesto) se deben expulsar mediante un archivo de parámetros LCM, como el que se muestra en el siguiente ejemplo.

```
Options
  NoSync
;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Eject
```

```
When(
  (InLsm)
  and
  (DataSetName EQ 'DRVLT.MANIFEST')
  and
  (TMSScratch EQ False)
)
Control(Serial)
Ejmsg('Move to DR Vault')
;
```

En este ejemplo:

- La sentencia *OPTIONS* especifica *NOSYNC*, porque el almacenamiento no usa información de TMS para almacenamiento.
- La sentencia *VAULT* especifica *DRVLT* como el almacén de DR.
- La sentencia *ACTION EJECT* especifica:
 - Expulsar la cinta del archivo de manifiesto.
 - Expulsar los volúmenes nativos no reutilizados por el TMS.
 - El mensaje de expulsión.

Paso 6: Creación de una lista desplegable de volúmenes para devolución a partir del almacén

Para crear una lista desplegable, use un archivo de parámetros *LCM*, como el que se muestra en el siguiente ejemplo.

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Report
  Volume
  Sysout(*)
  Title('Return Report')
  When(
    (VaultName EQ 'DRVLT')
    and
    (VaultReturnDate LE TODAY)
    and
    (VaultReturnDate NE MISSING)
  )
  Column (Serial,
          VaultSlot)
;
```

En este ejemplo:

- La sentencia *OPTIONS* especifica *NOSYNC* y *NOTMS*, porque el almacenamiento no usa información de TMS para almacenamiento y no se necesitan metadatos de TMS.
- La sentencia *VAULT* especifica *DRVLT* como el almacén de DR.
- La sentencia *REPORT VOLUME* crea un informe que muestra los volúmenes del almacén que han llegado a las fechas de devolución asignadas anteriormente. Este es un ejemplo simple; puede agregar más criterios de selección para los volúmenes que desea que se devuelvan.

Nota:

- *TODAY* y *MISSING* son valores únicos en relación a las fechas. *TODAY* se convierte a la fecha de ejecución de LCM. *MISSING* significa que no hay ningún valor para la fecha. En este ejemplo, significa que no se ha establecido ninguna fecha. Ambas condiciones son necesarias, dado que una fecha faltante se vería como *Less Than* respecto de la fecha actual.
 - Los pasos 4, 5 y 6 se pueden combinar en un solo paso de trabajo. En algunos casos, el paso 6 se ejecuta periódicamente, por lo general, el día antes de que los volúmenes se devuelvan del almacén.
-

Paso 7: preparación de MVC almacenados para devolución

Los MVC se preparan para devolución mediante un archivo de parámetros *LCM*, como el que se muestra en el siguiente ejemplo.

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Drain
  When(
    (MVC EQ True)
    and
    (VaultName EQ 'DRVLT')
    and
    (MVCVTVCount LE 30)
    and
    (MVCInUse LE 30)
    and
    (Days_Since(VaultAssignmentDate) GT 7)
  )
  Control(MVCVTVCount
    Ascending)
  Limit(30)
;
```

En este ejemplo:

- La sentencia *OPTIONS* especifica *NOSYNC* y *NOTMS*, porque el almacenamiento no usa información de TMS para almacenamiento y no se necesitan metadatos de TMS.

- La sentencia *VAULT* especifica *DRVLT* como el almacén de DR.
- Para los MVC que actualmente están en el almacén de DR, la sentencia *ACTION DRAIN* especifica el drenaje de los MVC:
 - Que tengan menos de 30 VTV.
 - Que estén en uso menos del 30 % del tiempo.
 - Que hayan estado en el almacén al menos 7 días.
 - Que limiten la cantidad de MVC devueltos a un máximo de 30.
 - El parámetro *GracePeriod* establece la fecha de devolución en tres días.

Cuando crea un archivo de parámetros para drenar los MVC, necesita equilibrar los ciclos de procesamiento para el drenaje y la necesidad de reciclar MVC fragmentados o parcialmente llenos. Por lo tanto, los parámetros *MVCVTVCount*, *MVCInUse*, *Days_Since* y *LIMIT* proporcionan los controles para equilibrar estas necesidades.

Paso 8: Preparación de volúmenes nativos almacenados para devolución

Los volúmenes nativos se preparan para devolución mediante un archivo de parámetros *LCM*, como el que se muestra en el siguiente ejemplo.

```
Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
  ;
Vault
  Name('DRVLT')
  GracePeriod(3)
  ;
Action
  Vault
  Return
  When(
  Not (MVC)
  and
  (VaultName EQ 'DRVLT')
  and
  (TMSScratch EQ True)
  )
  ;
```

En este ejemplo:

- La sentencia *OPTIONS* especifica *NOSYNC*, porque el almacenamiento no usa información de TMS para almacenamiento.
- La sentencia *TMS RMM* es necesaria para agregar el procesamiento de metadatos de TMS.
- La sentencia *VAULT* especifica *DRVLT* como el almacén de DR.

- La sentencia *ACTION VAULT RETURN* establece una fecha de devolución (mediante el parámetro *GracePeriod*) para volúmenes que no son MVC y que están en estado reutilizable en el TMS.

Paso 9: Introducción de volúmenes devueltos

Los volúmenes que aparecen en el informe creado en "[Paso 6: Creación de una lista desplegable de volúmenes para devolución a partir del almacén](#)" se eliminan del almacén de DR y se devuelven al entorno local. Cuando introduce estos volúmenes en el ACS, el HSC verifica si la fecha de devolución del almacén asignada para cada volumen ya ha pasado. Si ya ha pasado, significa que el volumen ha alcanzado la fecha de devolución programada y se elimina del almacén. A continuación, los MVC devueltos se convierten en elegibles para migración, y los volúmenes nativos se convierten en reutilizables en el CDS la próxima vez que se produce el procesamiento de SYNC del LCM. Si la fecha de devolución del almacén no ha pasado, el volumen se expulsa mediante "[Paso 4: Expulsión de los MVC de almacén](#)."

Si lo desea, puede combinar los pasos 7 y 8 en un solo archivo de parámetros *LCM*, como se muestra en el siguiente ejemplo:

```
Options
  NoSync
;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Drain
  When(
    (MVC EQ True)
    and
    (VaultName EQ 'DRVLT')
    and
    (MVCVTVCount LE 30)
    and
    (MVCInUse LE 30)
    and
    (Days_Since(VaultAssignmentDate) GT 7)
  )
  Control(MVCVTVCount
    Ascending)
  Limit(30)
;
Action
  Vault
  Return
  When(
    Not (MVC)
    and
    (VaultName EQ 'DRVLT')
    and
    (TMSScratch EQ True)
```

)
;

Almacenamiento de DR de varias semanas con MVC

Algunos sitios pueden optar por usar un proceso de varias semanas en el que el procesamiento de DR incluya una copia de seguridad semanal y completa de volúmenes de datos críticos, seguida de copias de seguridad incrementales durante los seis días siguientes. Mediante el proceso de almacenamiento externo se mueven los volúmenes fuera del sitio el día de la creación. Este proceso supone un ciclo de cuatro semanas durante el cual los datos de DR se mantienen completos y fuera del sitio hasta que el comienzo de la cuarta semana. Se debe tener en cuenta que el único cambio entre "[Almacenamiento básico de DR con MVC](#)" y el proceso de varias semanas es la diferencia en el criterio de selección para los pasos 7 y 8, como se describe a continuación. Esta acción se realiza configurando las fechas de caducidad, de modo que los VTV de almacén en los MVC asociados y las cintas del archivo de manifiesto (y todas las demás cintas nativas involucradas) caduquen en la fecha especificada 22 días después de la creación.

La cronología de varias semanas se desarrolla como se indica a continuación:

- Día 1: copias de seguridad de volúmenes completas (caducan el vigésimo segundo día).
- Día 2: copia de seguridad incremental n.º 1 (caduca el vigésimo segundo día).
- Día 3: copia de seguridad incremental n.º 2 (caduca el vigésimo segundo día).
- Día 4: copia de seguridad incremental n.º 3 (caduca el vigésimo segundo día).
- Día 5: copia de seguridad incremental n.º 4 (caduca el vigésimo segundo día).
- Día 6: copia de seguridad incremental n.º 5 (caduca el vigésimo segundo día).
- Día 7: copia de seguridad incremental n.º 6 (caduca el vigésimo segundo día).
- Días 8 a 21: los volúmenes permanecen fuera del sitio.
- Día 22: las copias de seguridad y las cintas del archivo de manifiesto de los días 1 a 7 caducan, y los VTV se reutilizan en CDS mediante el proceso LCM VTVSYNC. Los MVC se drenan mediante el siguiente parámetro en el criterio de selección para "[Paso 7: preparación de MVC almacenados para devolución](#)" y "[Paso 8: Preparación de volúmenes nativos almacenados para devolución](#)."

DAYS SINCE (VaultAssignmentDate) GT 15

La fecha de devolución para los MVC drenados y las cintas del archivo de manifiesto se establece para el vigésimo quinto día.

Nota:

Todos los MVC asignados al almacén durante los primeros siete días del ciclo se drenan. Si la fecha de caducidad de los VTV se estableció correctamente, no debe haber VTV actuales en este momento y el proceso de drenaje lógico se debe ejecutar rápidamente. La escritura de VTV actuales en un nuevo MVC se podría producir debido al establecimiento de una fecha de caducidad incorrecta.

- Días 23 y 24: los volúmenes permanecen fuera del sitio.
- Día 25: los volúmenes almacenados del día 1 al 7 se devuelven y se eliminan del estado almacenado, y quedan disponibles para reutilización.
- Día 29: se repite el ciclo.

Nota:

Algunos sitios pueden optar por dejar las copias de seguridad de volúmenes completas fuera del sitio y por mantener las copias de seguridad incrementales en el sitio. En este caso, los MVC incrementales, el archivo de manifiesto relacionado y los volúmenes nativos se deben colocar en un contenedor que se debe cerrar con llave y no se debe volver a abrir hasta su devolución el día 25. Para ese momento, todos los volúmenes creados el día 1 deben haber caducado.

Almacenamiento de DR con MVC en una biblioteca remota

En este proceso, en lugar de almacenar volúmenes en un almacén físico, los volúmenes se almacenan en una biblioteca remota (ACS). Este proceso es similar al proceso que se describe en "[Almacenamiento básico de DR con MVC](#)", con las siguientes excepciones:

- Pasos 1 a 3: sin cambios.
- Paso 4: se elimina porque no se realizan expulsiones de MVC almacenados.
- Paso 5: al igual que en el paso 4, los volúmenes nativos no se expulsan. En cambio, una sentencia *Action Vault Assign* asigna volúmenes nativos al almacén mediante los mismos criterios de selección que se hubieran utilizado para la realización y la expulsión de estos volúmenes nativos, como se muestra en el siguiente ejemplo.

```
Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
  ;
Vault
  Name('DRVLT')
  GracePeriod(3)
  ;
Action
  Vault
  Assign
  Vault('DRVLT')
  when(
  (InLsm)
  and
  (DataSetName EQ 'DRVLT.MANIFEST'))
```

```
and
(TMSScratch EQ False)
)
;
```

- Paso 6: se elimina porque no se vuelven a introducir volúmenes.
- Paso 7: sin cambios.

Nota:

Dado que el procesamiento del drenaje se produce en la biblioteca remota, los drenajes se ejecutan más eficazmente que el procesamiento de drenaje lógico para los volúmenes de un almacén manual.

- Paso 8: sin cambios.
- Paso 9: en el proceso básico, la asignación del almacén se elimina mediante el proceso de introducción, que no ocurre en el escenario de la biblioteca remota. Para eliminar el volumen almacenado, use la sentencia *Action Vault Release*. Los volúmenes almacenados que se liberarán se seleccionan por *Vault Name* y *Return Date*, como se realizó anteriormente para generar la lista desplegable. La sentencia *Action Vault Release* solo procesa volúmenes almacenados cuya *Return Date* ya haya pasado.

A continuación, se muestra un ejemplo de la sentencia *Action Vault Release*.

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Vault
  Release
  When(
    (VaultName EQ 'DRVAULT')
    and
    (VaultReturnDate LE TODAY)
    and
    (VaultReturnDate NE MISSING)
  )
;
```

Almacenamiento de MVC para LTR

El proceso para usar los MVC para alojar MVC de retención a largo plazo (LTR) es básicamente el mismo que se describe en el proceso básico para recuperación ante desastres,

en " [Almacenamiento básico de DR con MVC](#)". Las dos consideraciones principales para el uso de LTR son las siguientes:

- El movimiento al almacén, "[Paso 2: Exportación de los MVC de almacén](#)" a "[Paso 5: Expulsión de volúmenes nativos \(incluida la cinta del archivo de manifiesto\)](#)," puede no ser una función diaria, de modo que más volúmenes de LTR se llenen completamente antes de ser almacenados.
- Los VTV de LTR no caducan por períodos de tiempo prolongados; por lo tanto, el proceso periódico descrito puede producirse únicamente a intervalos más prolongados. Seguirán existiendo MVC de LTR que estén solo parcialmente llenos; por lo tanto, en algún momento, los MVC parcialmente llenos con menos VTV y bajo uso de MVC se deberán procesar para que los VTV de los MVC parcialmente llenos se consoliden en menos MVC de LTR.

En algún momento del futuro, puede existir la necesidad de realizar drenajes lógicos de los MVC de LTR para mover los datos archivados a nuevos medios. El proceso básico realiza fácilmente esa actividad, simplemente, mediante las elecciones apropiadas de los criterios de selección y la limitación de la cantidad de MVC de LTR procesados a la vez. Con cada ejecución, los VTV actuales se mueven a los medios nuevos, los MVC se mueven al almacén y los MVC drenados lógicamente se devuelven para reutilización o destrucción.

Expulsión de volúmenes específicos a un almacén local (planta)

Es posible que muchos sitios requieran la eliminación de volúmenes específicos del entorno automatizado y que necesiten almacenar eficazmente estos volúmenes localmente en bastidores del entorno local. Esta necesidad puede surgir a partir de actividades tales como retirar los volúmenes del uso y a partir del deseo de mantener los volúmenes disponibles durante algún tiempo. Varios almacenes locales (planta) se pueden definir con volúmenes específicos que se envían a distintos almacenes, pero solo un almacén local (planta) se puede definir como un almacén "predeterminado". A todos los volúmenes colocados en un almacén local/planta se les asigna automáticamente una fecha de devolución a partir la fecha actual. Esto permite que estos volúmenes se devuelvan al entorno automatizado y se supriman de la asignación de su almacén sin necesidad de realizar ninguna otra acción.

En el siguiente ejemplo, se muestra un ejemplo de un archivo de parámetros LCM mediante el cual se expulsan volúmenes específicos a un almacén de planta.

```
Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
  ;
Vault
  Name('FLOOR')
```

```
    Default
    ;
Action
  Eject
  When(
    (InLsm EQ True)
    and
    (DaysSinceReference GT 100)
    and
    (MVC EQ False)
    and
    Not
    (DataSetName Matches 'HMIG.**')
  )
  Control(
    VaultSlot
    Ascending
  )
  Ejmsg('Move to Floor Vault')
  ;
Manage
  ACSID(00)
  Numfree(500)
  ;
```

En este ejemplo:

- La sentencia *OPTIONS* especifica *NOSYNCH*, porque el almacenamiento no usa información de TMS para almacenamiento.
- La sentencia *TMS RMM* es necesaria para agregar el procesamiento de metadatos de TMS.
- La sentencia *VAULT* especifica *FLOOR* como el almacén de planta predeterminado.
- La sentencia *ACTION EJECT* especifica la expulsión al almacén de planta de los volúmenes:
 - Que estén en el LSM.
 - Que no se hayan consultado durante más de 100 días.
 - Que *no* sean MVC.
 - Que tengan una máscara de nombre de juego de datos *HMIG.***.
- La sentencia *ACTION EJECT* también especifica:
 - Procesar los volúmenes en orden de volser ascendente.
 - Expulsar por números de ranura de TMS.
 - El mensaje de expulsión.
- La sentencia *MANAGE* especifica:
 - Gestionar volúmenes en ACS 00.
 - Garantizar 500 celdas libres en el ACS.

Capítulo 4. Uso de replicación cruzada entre sistemas TapePlex en una solución de DR

En el "[Realización de exportaciones e importaciones físicas](#)", se explica cómo crear MVC portátiles de "exportación" a partir de un sitio de origen, de modo de poder mover físicamente estos MVC a un sitio de destino e importar los MVC (y los VTV que contienen) en el sitio de destino. Con la replicación cruzada entre sistemas TapePlex (CTR), ya no deberá usar el PTAM (Pickup Truck Access Method) para mover los MVC de un sitio a otro. En cambio, mueve los VTV electrónicamente desde el sitio de origen hasta el de destino, es decir, de un TapePlex a otro, donde los VTV, luego, se migran a los MVC y, de esa manera, se elimina el paso de PTAM. Dado que una copia del VTV se mueve del TapePlex de origen al de destino, una copia de los metadatos del VTV se mueve del CDS del TapePlex de origen al CDS del TapePlex de destino. Tenga en cuenta que el TapePlex de origen sigue siendo el "propietario" y sigue gestionando la reutilización de los VTV de CTR.

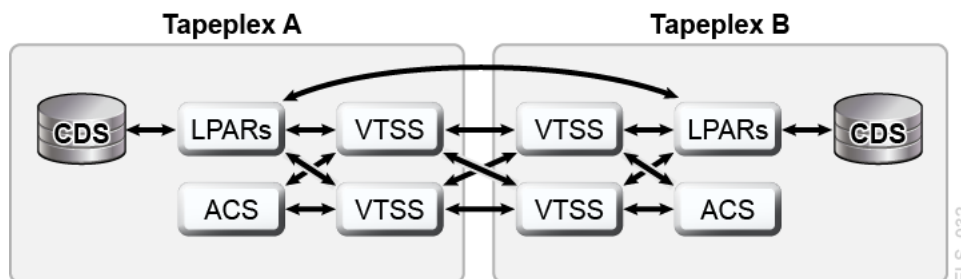
Atención:

Tenga en cuenta que si está utilizando CTR, al detener el SMC, se impide que el VTCS envíe metadatos a un TapePlex de CTR, lo cual detiene eficazmente la transferencia de datos. Por lo tanto, si está usando una función de HSC que usa los servicios de comunicación de SMC, como CTR, debe asegurarse de que la actividad de HSC se desactive o que se termine el HSC antes de detener el SMC.

Funcionamiento de CTR

Observe una CTR, como se muestra en [Figura 4.1, "Configuración de CTR de ELS"](#).

Figura 4.1. Configuración de CTR de ELS



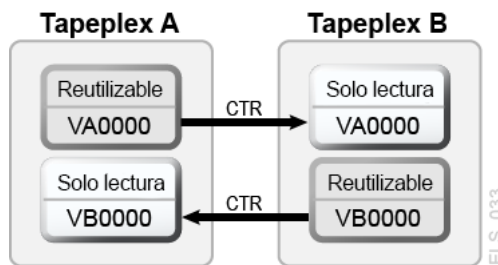
Como se muestra en [Figura 4.1, "Configuración de CTR de ELS"](#):

- CTR usa la conexión entre dos VTSS (CLINK) en dos TapePlex independientes para enviar datos de un VTSS a otro. La conexión puede ser unidireccional o bidireccional.
- CTR usa los servicios de la función de cliente/servidor de SMC para enviar metadatos del TapePlex emisor al TapePlex receptor. Tenga en cuenta que NO es necesario usar la función de cliente/servidor para comunicarse entre el SMC y el HSC para usar CTR, pero debe definir los comandos HTTP y SERVER en SMC para permitir la transferencia de los metadatos.
- Hay CDS separados (y que se mantienen de forma separada) en cada sitio, de modo que la pérdida de conectividad o disponibilidad de hardware de un sitio no afecte directamente a cualquiera de los demás sitios.
- Los requisitos de configuración y conexión física son simples.
- Ahora, puede ejecutar pruebas de DR concurrente de una manera más sencilla y sin interrupciones para el trabajo existente (sin usar la utilidad CDRT).
- Ahora, puede hacer un switchover automático de la carga de trabajo de un sitio a otro.
- Los rangos de volúmenes de VTV para los dos TapePlex se muestran en [Figura 4.2, "Relaciones de volúmenes de VTV dentro del sitio"](#). **Tenga en cuenta que** cada TapePlex tiene su propio juego de volúmenes de escritura y que las versiones de solo lectura los reflejan en el otro TapePlex.
- La configuración que se muestra tiene ambos VTSS en el TapePlex emisor conectado a ambos VTSS en el TapePlex receptor para máxima capacidad de recuperación.

Nota:

Tanto en las configuraciones de VTSS en cluster como en las de CTR, debe asegurarse de que los primeros 16 VTD de cada VTSS (0-F) se reserven para replicación. Estos dispositivos deben estar FUERA DE LÍNEA para los MVS, y sus rutas deben estar en línea para cada host de servidor de HSC. VTCS no registra los primeros 16 VTD con SMC/HSC, lo cual impide montar los VTV en estos VTD.

Figura 4.2. Relaciones de volúmenes de VTV dentro del sitio



Lea "[Consideraciones de VTV de CRT de solo lectura](#)," y, luego, vaya a "[Configuración para CTR](#)."

Consideraciones de VTV de CRT de solo lectura

Cuando usa CTR, todos los VTV replicados de un sitio a otro se encuentran en modo de solo lectura en el sitio remoto. Si bien estos VTV pueden ser reutilizados (y sus respectivos

números de serie de volumen se pueden reutilizar) por el TapePlex remoto en caso de un desastre real, el estado de solo lectura no se puede modificar mientras los volúmenes no estén en un estado SCRATCH. Tenga en cuenta que los volúmenes de una agrupación *POOLPARAM EXTERNAL* nunca se pueden establecer en un estado SCRATCH.

Por lo tanto, si decide usar CTR como estrategia de continuidad empresarial o recuperación ante desastres, debe asegurarse de que las aplicaciones no intenten actualizar estos volúmenes, ya sea durante una prueba de DR o ante un desastre real. Se deben tener en cuenta los siguientes escenarios:

1. Las aplicaciones que usan el atributo *DISP=MOD* en JCL o una asignación dinámica para agregar datos a un juego de datos existente deben implementar un mecanismo de punto de control/reinicio y deben registrar un punto de control antes de crear volúmenes *DISP=MOD*. Estas aplicaciones se recuperan al reiniciarse en el punto de control y, si corresponde, deben volver a crear estos volúmenes *DISP=MOD* una vez reiniciados. Tenga en cuenta que el uso de *DISP=MOD* en sí no es un problema para la replicación cruzada entre sistemas TapePlex. Mientras la aplicación tenga un punto de control que le permita retirar actualizaciones parciales o un diseño que permita que la salida de nuevos datos comience en un nuevo volumen, se debe poder ejecutar en VTV de solo lectura sin problemas.
2. Si los VTV que se replican en otro TapePlex le pertenecen al HSM, el siguiente proceso permite que la recopilación de datos comience en un nuevo volumen y evita la actualización de VTV de HSM existentes:
 - a. Marque los volúmenes existentes como completos.
 - b. Modifique *ARCCMD* si es necesario para *USERUNITTABLE*, *MIGRATION*, *BACKUP* y *RECYCLE*.
 - c. Asegúrese de que *RECYCLEDALLOCFREQ* esté establecido en 1. Esto permitirá a la asignación de HSM asignar un nuevo volumen y un nuevo dispositivo cuando corresponda.
 - d. En función de su *MGMTCLAS* *VTVSIZE*, establezca *PERCENTFULL*:

Para VTV de 800 MB, establezca *PERCENTFULL* de HSM en 97.

Para VTV de 4 GB, establezca *PERCENTFULL* de HSM en 450.

Para VTV de 32 GB, establezca *PERCENTFULL* de HSC en 3600.

Las aplicaciones que apilan juegos de datos en un volumen existente están sujetas a las mismas restricciones de *DISP=MOD* que las que se indican anteriormente.

Configuración para CTR

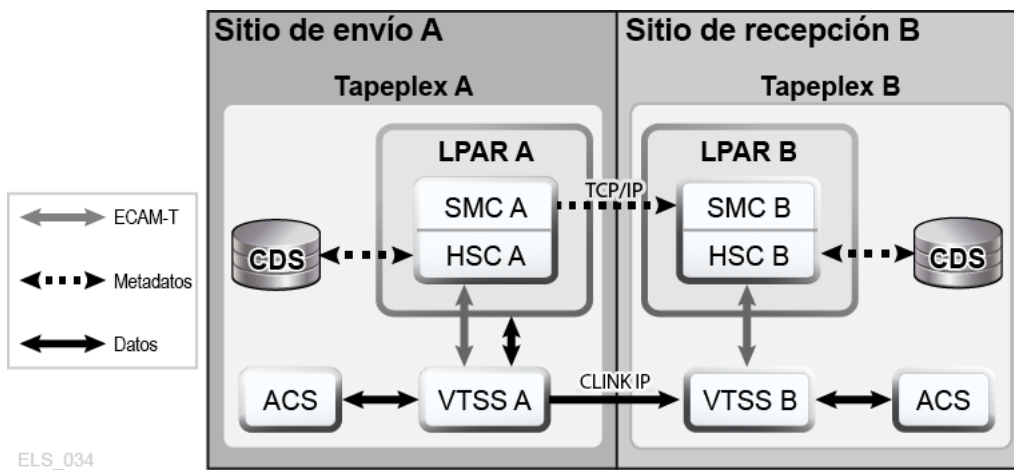
En [Figura 4.3, “Configuración de CTR”](#), se muestra un ejemplo de configuración de CTR. En este sistema, el VTSSA del VTSS reside en el TapePlex TAPEPLXA y tiene CLINK "asociados" al VTSSB del VTSS en el TapePlex TAPEPLXB. Los VTV replicados en

VTSSB ahora residen en el CDS de TAPEPLXB, como los MVC a los que los VTV se migran posteriormente. Es decir, los VTV se replican en sistemas TapePlex y, luego, se migran localmente. Los VTSS que se encuentran en el TapePlex emisor no pueden tener conexiones con las RTD del TapePlex receptor.

Nota:

En el siguiente ejemplo, se muestra una CTR unidireccional. Para hacer una CTR bidireccional, simplemente, debe definir la configuración y las sentencias de control de cliente/servidor de SMC de la misma manera que en ambos sistemas TapePlex. Tenga en cuenta que un TapePlex único también puede recibir VTV de varios sistemas TapePlex. Para definir una configuración donde un TapePlex recibe datos de varios sistemas TapePlex, simplemente, debe agregar nombres adicionales de TapePlex a *CONFIG* of *TAPEPLXB*.

Figura 4.3. Configuración de CTR



Instalación: configuración e inicio de CTR

Para configurar e iniciar el ejemplo de sistema CTR que se muestra en "[Configuración para CTR](#)" haga lo siguiente:

1. Asegúrese de que su sistema cumpla con los requisitos de VTSS en cluster descritos en *Instalación de ELS*.
2. Inicie el servidor de HTTP en el SMC que se ejecuta en la LPARB host.

Se recomienda hacer esto en el archivo CMDS de SMC. Por ejemplo:

```
HTTP START PORT(999)
```

3. Defina los comandos TAPEPLEX y SERVER en la LPARB host.

Nuevamente, se recomienda hacer esto en el archivo CMDS de SMC. Por ejemplo:

```
TAPEPLEX NAME(TAPEPLXA) LOCSUB(HSCA)
```

```
TAPEPLEX NAME(TAPEPLXB)
SERVER NAME(REMB)TAPEPLEX(TAPEPLXB) HOSTNAME(LPARB) PORT(999)
```

Nota:

En el ejemplo de configuración, el TAPEPLXB de TapePlex existe (desde la perspectiva del TAPEPLXA de TapePlex) con la única finalidad de mantener un CDS que contenga metadatos de VTV que se replicaron a partir de TAPEPLXA. No obstante, si las definiciones de HSC o VTCS en los sistemas TapePlex TAPEPLXA y TAPEPLXB usan las mismas direcciones de dispositivos para hacer referencia a distintos dispositivos físicos, debe definir comandos *SMC UNITATTR* para indicarle al SMC qué TapePlex define los dispositivos en su host. Si bien *UNITATTR* debe especificar un *MODEL*, si el modelo especificado no coincide con el modelo informado por el TapePlex, el modelo real sustituye al *UNITATTR MODEL*. A continuación, se muestra un ejemplo de la sentencia *UNITATTR* de SMC que se utilizaría si los sistemas TapePlex TAPEPLXA y TAPEPLXB definieran el rango de direcciones 9000-90FF:

```
UNITATTR ADDR(9000-90FF) TAPEPLEX(TAPEPLXA) MODEL(VIRTUAL)
```

4. Codifique una plataforma *CONFIG* para TapePlex A, como se muestra en [Ejemplo 4.1, “Plataforma CONFIG para TapePlex A”](#).

En esta figura, tenga en cuenta lo siguiente:

- La sentencia *TAPEPLEX*, que define este TapePlex.
- Las sentencias *CLINK* definen los CLINK que se utilizan para CTR de VTSSA a VTSSB.
- La configuración de replicación condicional en la sentencia *CONFIG GLOBAL* es *CHANGED* para TAPEPLXA.

5. Codifique una plataforma *CONFIG* para TapePlex B, como se muestra en [Ejemplo 4.2, “Plataforma CONFIG para TapePlex B”](#).

En esta figura, tenga en cuenta lo siguiente:

- La sentencia *TAPEPLEX* incluye un parámetro *RECVPLEX=TAPEPLXA* para especificar que TAPEPLXB puede recibir VTV de TAPEPLXA.
- No hay sentencias *CLINK*, porque los CLINK se definen en la plataforma *CONFIG* para TAPEPLXA.

Ejemplo 4.1. Plataforma CONFIG para TapePlex A

```
//CREATCFG EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCNTL DD DSN=h1q.TAPEPLXA.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=h1q.TAPEPLXA.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=h1q.TAPEPLXA.DBASESBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
```

```

GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTPAGE=LARGE REPLICAT=CHANGED
RECLAIM THRESHLD=70 MAXMVC=30 START=98 CONMVC=1
TAPEPLEX THISPLEX=TAPEPLXA
VTSS NAME=VTSSA LOW=71 HIGH=80 MAXMIG=8 MINMIG=1 RETAIN=10
RTD NAME=VSMA1A00 DEVNO=1A00 CHANIF=0C
RTD NAME=VSMA1A01 DEVNO=1A01 CHANIF=0D
RTD NAME=VSMA1A02 DEVNO=1A02 CHANIF=0K
RTD NAME=VSMA1A03 DEVNO=1A03 CHANIF=0L
RTD NAME=VSMA2A08 DEVNO=2A08 CHANIF=1C
RTD NAME=VSMA2A09 DEVNO=2A09 CHANIF=1D
RTD NAME=VSMA2A0A DEVNO=2A0A CHANIF=1K
RTD NAME=VSMA2A0B DEVNO=2A0B CHANIF=1L
VTD LOW=7900 HIGH=79FF
VTD LOW=8900 HIGH=89FF
CLINK VTSS=VTSSA CHANIF=0G REMPLEX=TAPEPLXB PARTNER=VTSSB
CLINK VTSS=VTSSA CHANIF=00 REMPLEX=TAPEPLXB PARTNER=VTSSB

```

Ejemplo 4.2. Plataforma CONFIG para TapePlex B

```

//CREATCFG EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLS_CNTL DD DSN=h1q.TAPEPLXB.DBASEPRM,DISP=SHR
//SLS_CNTL2 DD DSN=h1q.TAPEPLXB.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=h1q.TAPEPLXB.DBASESBY,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=98 CONMVC=1
TAPEPLEX THISPLEX=TAPEPLXB RECVPLEX=TAPEPLXA
VTSS NAME=VTSSB LOW=75 HIGH=80 MAXMIG=8 MINMIG=1 RETAIN=10
RTD NAME=VSMB3A00 DEVNO=3A00 CHANIF=0C
RTD NAME=VSMB3A01 DEVNO=3A01 CHANIF=0D
RTD NAME=VSMB3A02 DEVNO=3A02 CHANIF=0K
RTD NAME=VSMB3A03 DEVNO=3A03 CHANIF=0L
RTD NAME=VSMB4A08 DEVNO=4A08 CHANIF=1C
RTD NAME=VSMB4A09 DEVNO=4A09 CHANIF=1D
RTD NAME=VSMB4A0A DEVNO=4A0A CHANIF=1K
RTD NAME=VSMB4A0B DEVNO=4A0B CHANIF=1L

```

Definición de políticas para CTR

Consulte los siguientes procedimientos para definir políticas para CTR.

Políticas para el TapePlex emisor

Para definir políticas para el TapePlex emisor (TAPEPLXA) del ejemplo de sistema CTR que se muestra en [Figura 4.3, “Configuración de CTR”](#), haga lo siguiente:

1. Cree definiciones *MVC POOLPARAM/VOLPARAM* para TAPEPLXA:

```
POOLPARAM TYPE(MVC) NAME(MVCPLA) INITMVC(YES) MVCFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARAM VOLSER(AM1000-AM1299) MEDIA(STK1R)
```

2. Cree definiciones de agrupaciones reutilizables *VTV POOLPARAM/VOLPARAM* para TAPEPLXA:

```
POOLPARAM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL) REC(VIRTUAL)
```

3. Para TAPEPLXA, cree las clases de almacenamiento para los MVC que contengan los VTV migrados lógicamente y las clases de almacenamiento de CTR.

```
STOR NAME(LOCAL1) ACS(00) MEDIA(STK1R)
STOR NAME(EIPA1) TAPEPLEX(TAPEPLXB)
```

En el ejemplo anterior, las sentencias *STORClas* definen:

- La clase de almacenamiento LOCAL1, que es la clase de almacenamiento para los VTV migrados lógicamente de cada VTSS.
- La clase de almacenamiento EIPA1, que es la clase de almacenamiento de CTR y especifica el TapePlex receptor (TAPEPLXB).

4. Cree la clase de gestión que apunta a las clases de almacenamiento del paso 3.

```
MGMT NAME(LOCEEX1) MIGPOL(LOCAL1) EEXPOL(EIPA1)
```

5. Cree una política del SMC que especifique medios virtuales y asigne la clase de gestión creada en el paso 4.

```
POLICY NAME(PPAY) MEDIA(VIRTUAL) MGMT(LOCEEX1)
```

6. Cree una sentencia *TAPEREQ* para enrutar datos críticos a VSM y asignar la política correspondiente a los datos.

```
TAPEREQ DSN(*.PAYROLL.***) POLICY(PPAY)
```

En el ejemplo anterior, la sentencia *TAPEREQ* especifica que se deben enrutar los juegos de datos con la máscara de HLQ **.PAYROLL.*** a VSM y asignar una política *PPAY*.

Nota:

- Si bien puede usar políticas del SMC para dirigir los CTR a un grupo esotérico específico, StorageTek recomienda usar solo *MGMTCLAS*, de modo que la influencia de asignación de SMC/VTCS pueda usar cualquier VTSS que cumpla con los requisitos de *MGMTCLAS*.
 - Puede usar el comando *EEXPORT* para una CTR manual. Para obtener más información, consulte la *Referencia de comandos, sentencia de control y utilidades de ELS*.
-

7. Compruebe el miembro *SYS1.PARMLIB SMFPRMxx* para garantizar que se activen los registros del subtipo 28.

Si se activan, VTSS escribe un subtipo 28 que incluye el nombre del VTSS de destino para cada evento de CTR. Usted sabrá que la CTR se ha realizado correctamente cuando los VTV lleguen correctamente al VTSS del sitio receptor. ¿Hay alguna manera de confirmar esto? Sí. Use la utilidad *DRMONitr*, como se describe en el paso 8.

8. Cree un JCL para supervisar la CTR.

Para esto, use la utilidad *DRMONitr* para supervisar la CTR. *DRMONitr* hace que el trabajo de MVS asociado pause hasta que la CTR finalice correctamente. Por ejemplo:

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
/* If HSC IS NOT OR MAY NOT BE ACTIVE, INCLUDE THE
/* FOLLOWING:
//SLSCNTL DD DSN=primary.cds.name, DISP=SHR
//SLSCNTL2 DD DSN=secondary.cds.name, DISP=SHR
//SLSSTBY DD DSN=standby.cds.name, DISP=SHR
//SLSPARMP DD DSN=h1q.PARMLIB(BKPCNTL), DISP=SHR
//SLSPARMS DD DSN=h1q.PARMLIB(BKPCNTL2), DISP=SHR
//SLSPARMB DD DSN=h1q.PARMLIB(BKPSTBY), DISP=SHR
//SYSIN DD UNIT=SYSDA, SPACE=(TRK,1)
/* THE FOLLOWING IS USED BY THE SNAPSHOT UTILITY:
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
DRMON MGMT(LOCEEX1) STOR(EIPA1) MAXAGE(24) TIMEOUT(120)
```

En este ejemplo, la utilidad *DRMON* supervisa las migraciones de la clase de almacenamiento *EIPA1*, que es la clase de almacenamiento de CTR especificada por la clase de gestión *LOCEEX1* en el paso 4. Asimismo, supervise únicamente los VTV que se actualizaron en las últimas 24 horas y genere un timeout para *DRMON* después de 120 minutos.

Políticas para el TapePlex receptor

Para definir políticas para el TapePlex receptor (TAPEPLXB) del ejemplo de sistema CTR que se muestra en " [Configuración para CTR](#)", haga lo siguiente:

1. Cree definiciones MVC POOLPARAM/VOLPARAM para la agrupación de MVC definida para el TapePlex TAPEPLXB para alojar VTV de CRT de TAPEPLXA:

```
POOLPARAM TYPE(MVC) NAME(AMVCCTR) INITMVC(YES) MCVFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARAM VOLSER(BM1000-BM1099) MEDIA(STK1R)
```

Nota:

StorageTek recomienda enfáticamente que use la característica POOLPARAM/VOLPARAM para garantizar que los rangos de volúmenes se reserven para volúmenes replicados por CTR en el sitio remoto.

2. Cree una agrupación de VTV externa para VTV exportados por TAPEPLXA:

```
POOLPARAM TYPE(EXTERNAL) NAME(AEXTBPL) OWNRPLEX(TAPEPLXA)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL) REC(VIRTUAL)
```

Nota:

En este ejemplo, no se definen agrupaciones para trabajo de producción en TAPEPLXB, sino solo agrupaciones utilizadas por TAPEPLXA. Si el trabajo de producción se debe ejecutar en TAPEPLXB, se necesitan definiciones POOLPARAM y VOLPARAM adicionales para las agrupaciones de MVC y reutilizables para el trabajo de TAPEPLXB.

3. Cree una agrupación reutilizable de VTV para que el TapePlex TAPEPLXB use para el trabajo de TAPEPLXA:

```
POOLPARAM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARAM VOLSER(BV1000-BV1999) MEDIA(VIRTUAL) REC(VIRTUAL)
```

4. Cree una agrupación de MVC para el TapePlex TAPEPLXB para alojar VTV de producción o prueba de DR de TAPEPLXA (en una situación de desastre):

```
POOLPARAM TYPE(MVC) NAME(AMVCDR) INITMVC(YES) MCVFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARAM VOLSER(BM2000-BM2099) MEDIA(STK1R)
```

5. Para TAPEPLXB, cree las clases de almacenamiento para migración local.

```
STOR NAME(TPEPLXA1) MCVPOOL(AMVCCTR)
STOR NAME(TPEPLXA2) MCVPOOL(AMVCDR)
```

En este ejemplo, las sentencias STORclas definen las clases de almacenamiento TPEPLXA1 y TPEPLXA2 para migración local. Los nombres de las clases de almacenamiento nos permiten separar este trabajo del trabajo local del TAPEPLXB.

6. Cree las clases de gestión que apuntan a las clases de almacenamiento del paso 5.

```
MGMT NAME(LOCEEX1) MIGPOL(TPEPLXA1)
MGMT NAME(LOCPLXA) MIGPOL(TPEPLXA2)
```

Tenga en cuenta que el nombre LOCEEX1 coincide con el nombre de la clase de gestión que se utiliza en TAPEPLXA (esta clase de gestión se especifica en los metadatos de VTV que se envían del VTSS de TAPEPLXA), pero el ejemplo hace referencia a la clase de almacenamiento para migración local. Las definiciones de las clases de gestión y almacenamiento de TAPEPLXB pueden usar cualquier parámetro, incluido EEXPOL, para replicarse a un tercer TapePlex. Además, debe crear otro MGMTCLAS, LOCPLXA, para ser utilizado para migración durante una prueba de DR de una carga de trabajo de TAPEPLXA.

Uso de CTR cuando el sitio remoto no tiene LPAR

En algunos entornos, solo un sitio tiene LPAR que realizan actividades de cinta, mientras que un segundo sitio contiene solo hardware de VTSS y biblioteca, pero no contiene LPAR de MVS. Es posible configurar este entorno, de modo que la CTR se pueda usar como mecanismo de DR y de prueba de DR.

Para hacer esto, debe:

1. Ejecutar la función de cliente/servidor de SMC en el entorno de producción, de modo de tener al menos una LPAR de producción que no ejecute HSC/VTCS.

De manera alternativa, puede ejecutar el TapePlex de DR en la misma LPAR que el TapePlex de producción mediante la función del modo MULT. Consulte *Configuración de HSC y VTCS* para obtener más información sobre el uso de esta capacidad.

En este ejemplo, el TapePlex de producción es TAPEPLXA.

2. Cree un nuevo CDS mediante la definición del hardware (bibliotecas y VTSS) en el sitio remoto.
3. Inicie un HSC/VTCS con el nuevo CDS en una LPAR (MVSX) que no esté ejecutando actualmente el HSC/VTCS de producción o en la LPAR donde decidió ejecutar varias copias del HSC/VTCS con la función del modo MULT.

Nota:

Por cuestiones de fiabilidad, se recomienda que ejecute dos instancias de HSC/VTCS que apunten a TAPEPLXB en dos LPAR distintas, de modo que si una instancia no está disponible, los metadatos de los VTV replicados mediante replicación cruzada entre sistemas TapePlex se puedan enviar a la segunda instancia.

El sistema es TapePlex TAPEPLXB.

4. Defina parámetros para el sistema SMC en MVSX mediante la definición de ambos sistemas TapePlex: TAPEPLXA y TAPEPLXB.

Cada sistema SMC del complejo debe definir tanto el TapePlex TAPEPLXA (el TapePlex de producción) como el TAPEPLXB (el TapePlex de DR). Para poder seguir replicando VTV durante una prueba de DR, debe definir un servidor para TapePlex TAPEPLXB que apunte al host del sitio remoto. Por ejemplo:

```
TAPEPLEX NAME(TAPEPLXA) LOCSUB(HSCA)
TAPEPLEX NAME(TAPEPLXB)
SERVER NAME(TPLXBPR) TAPEPLEX(TAPEPLXB) HOST(MVSX) PORT(999)
SERVER NAME(TPLXBDR) TAPEPLEX(TAPEPLXB) HOST(MVSXDR) PORT(1234)
```

Nota:

En este ejemplo, se supone que, si bien los nombres de LPAR (MVSX) pueden ser idénticos entre el sitio de producción y de DR, los dos sitios tienen nombres de host de TCP/IP únicos.

5. Defina las políticas de VTCS en TAPEPLXA para permitir la CTR en TAPEPLXB.

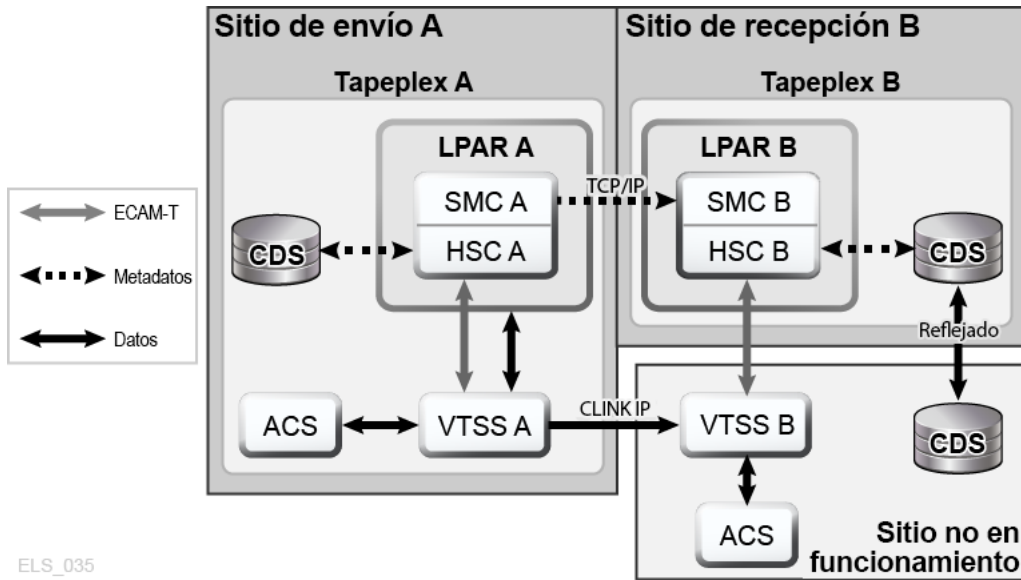
Consulte "[Definición de políticas para CTR.](#)"

6. Mediante la solución de replicación de discos, mantenga una copia del contenido del CDS para TAPEPLXB en la ubicación remota.

De manera alternativa, si existe una conectividad fiable, se recomienda que mantenga las copias principales (y otras) del CDS del HSC en el sitio de DR, mediante conexiones FICON para acceder al CDS desde el sitio de producción.

En [Figura 4.4, "Copia del CDS sin LPAR en el sitio remoto"](#), se muestra una copia del CDS sin LPAR en el sitio remoto.

Figura 4.4. Copia del CDS sin LPAR en el sitio remoto



Uso de CTR como solución de DR

Una solución de DR siempre le permite hacer tres cosas:

- Configurar e iniciar la solución como se describe en " [Instalación: configuración e inicio de CTR.](#)"
- Si ocurre un desastre, use la solución para continuar con las actividades empresariales en el sitio remoto, como se describe en " [Uso de CTR para continuidad empresarial.](#)"
- Use la solución para reanudar las actividades empresariales en el sitio local una vez que estén nuevamente en funcionamiento, como se describe en " [Uso de CTR para reanudación de las actividades empresariales.](#)"

Uso de CTR para continuidad empresarial

Si se produce una interrupción en el sitio TAPEPLXA, puede continuar trabajando en el sitio TAPEPLXB, simplemente, mediante la ejecución de la carga de trabajo con el TapePlex de TAPEPLXB. Para proteger los datos, los VTV que se replicaron a partir del TAPEPLXA permanecen en estado de solo lectura (consulte " [Consideraciones de VTV de CRT de solo lectura](#)"). No obstante, una vez que ha restablecido correctamente la carga de trabajo de TAPEPLXA, se recomienda que reutilice algunos VTV que se replicaron a partir de TAPEPLXA. Tenga en cuenta que antes de realizar este paso debe asegurarse de que el trabajo de producción de TAPEPLXA y TAPEPLXB se encuentre estable. Asimismo, en algún momento futuro, probablemente, deseará volver a crear un entorno independiente de TapePlex para que TAPEPLXA regrese a la configuración original.

Para usar CTR para continuidad empresarial:

1. Cambie las definiciones *POOLPARAM/VOLPARAM* en el CDS del TapePlex TAPEPLXB para la agrupación denominada (*AEXTBPL*) de *TYPE(EXTERNAL)* a *TYPE(SCRATCH)*:

```
POOLPARAM TYPE(SCRATCH) NAME(AEXTBPL)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL)
```

Tenga en cuenta que el rango de *VOLPARAM VOLSER* permanece sin cambios.

2. Ahora, puede ejecutar un trabajo de sincronización de volúmenes reutilizables en TAPEPLXB para reutilizar los VTV en el rango AV1000-AV1999, en función del estado reutilizable del TMS, ya sea para regresar a un punto de control o para realizar un procesamiento normal de actualización de volúmenes reutilizables.

Se recomienda dejar que pase un tiempo entre la acción de iniciar el procesamiento de la producción y permitir que los números de serie de volumen de VTV en el rango AV1000-AV1999 se reutilicen como volúmenes reutilizables. El uso de la característica *POOLPARAM/VOLPARAM* garantiza que estos volúmenes no se puedan seleccionar como reutilizables, a menos que una política solicite específicamente *SUBPOOL (AEXTBPL)*.

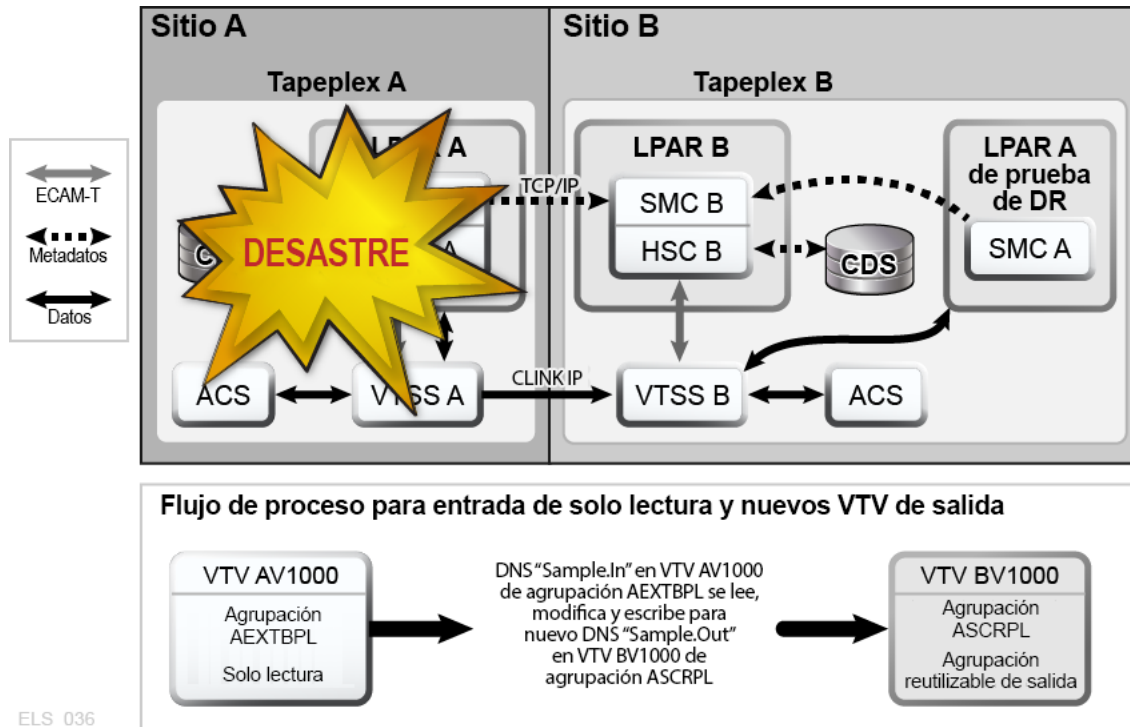
Durante este período usará volúmenes en el ASCRPL de la subagrupación reutilizable de TAPEPLXB (rango de volser: BV1000-BV1999) para el trabajo de producción de TAPEPLXA.

Una vez que el entorno de recuperación ante desastres esté estabilizado, podrá cambiar nuevamente las definiciones *POOLPARAM/VOLPARAM* para permitir la selección de volúmenes reutilizables a partir del rango AV1000-AV1999:

```
POOLPARAM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL)
VOLPARAM VOLSER(BV1000-BV1999) MEDIA(VIRTUAL)
```

En [Figura 4.5, “El sistema durante la continuidad empresarial”](#), se muestra un ejemplo de CTR con continuidad empresarial.

Figura 4.5. El sistema durante la continuidad empresarial



ELS_036

Uso de CTR para reanudación de las actividades empresariales

El sitio local experimentó una interrupción, usted siguió trabajando en el sitio remoto. Ahora, el sitio local está nuevamente en funcionamiento; por lo tanto, ¿cómo reanuda las actividades empresariales en el sitio local? Básicamente, la reanudación de las actividades depende de lo que haya ocurrido durante la interrupción y después de esta. Suponga que todos los datos locales originales se perdieron y que tiene un nuevo VTSS vacío en el sitio local.

Para reanudar las actividades después de perder todos los datos del sitio local:

1. Cree un nuevo CDS y ejecute una auditoría del HSC para determinar el contenido de las bibliotecas físicas.

A continuación, tendrá que hacer una "replicación invertida" de los datos y los metadatos en el sitio local a partir del sitio remoto.

2. Configure la plataforma CONFIG para el sitio remoto, de modo que pueda enviar datos al sitio local.
3. Realice la replicación invertida mediante EEXPORT.

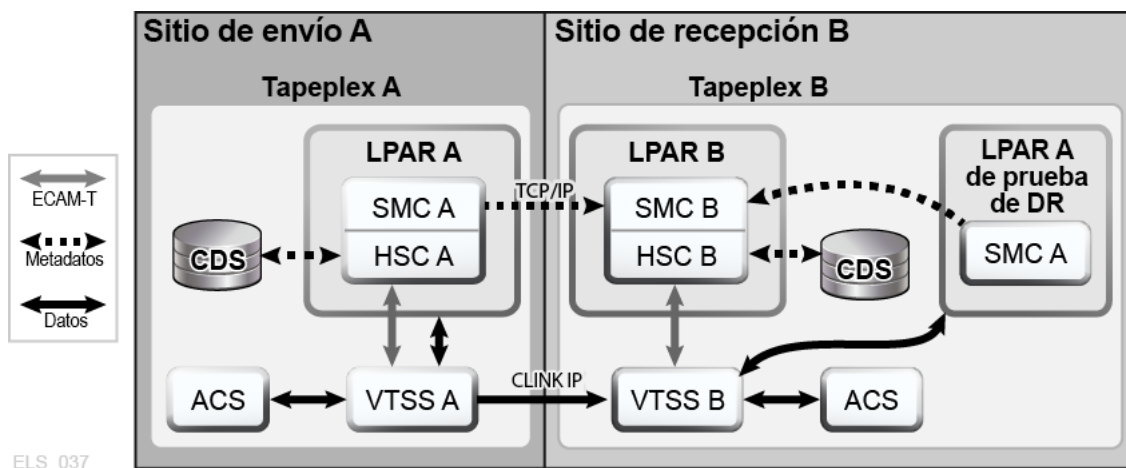
Por ejemplo:

```
EEXPORT MGMTCLAS(LOCEEX1, LOCEEX2) TOPLEX(TAPEPLXA)
```

Pruebas de recuperación ante desastres mediante la replicación cruzada entre sistemas TapePlex

Para continuar con el ejemplo, hay dos sitios, TAPEPLXA y TAPEPLXB, cada uno definido como su propio TapePlex (CDS del HSC), y usted ha usado la función de replicación cruzada entre sistemas TapePlex para replicar los VTV críticos de TAPEPLXA a TAPEPLXB. En [Figura 4.6, “Pruebas de recuperación ante desastres mediante la replicación cruzada entre sistemas TapePlex”](#), se muestra la realización de pruebas de DR con CTR.

Figura 4.6. Pruebas de recuperación ante desastres mediante la replicación cruzada entre sistemas TapePlex



Para realizar una prueba de DR en TAPEPLXB del trabajo de TAPEPLXA, se recomienda realizar el siguiente procedimiento:

1. Asegúrese de que el CDS de TAPEPLXB contenga una o más subagrupaciones reutilizables para los datos de salida de TAPEPLXA que estén separados de las subagrupaciones reutilizables que se utilizan para el trabajo de TAPEPLXB.

Para ver ejemplos, consulte " [Políticas para el TapePlex receptor.](#)"

2. Asegúrese de que los datos de catálogo y gestión de cintas de TAPEPLXA estén disponibles.
3. Abra el SMC en la LPAR de prueba de TAPEPLXA; para ello, defina el TapePlex como TAPEPLXB y especifique los comandos *SERVER* para uno o más hosts HSC en TAPEPLXB.
4. Antes de ejecutar la carga de trabajo de prueba.

El SMC tendrá acceso automáticamente a los VTV que existían antes o después del inicio de la prueba, porque estos se siguen replicando a partir de TAPEPLXA. Asegúrese de que el TapePlex TAPEPLXA no reutilice ni altere los VTV que usará la prueba de DR.

5. Cuando finalice la prueba, reutilice todos los VTV de las subagrupaciones de prueba de DR utilizados por la prueba.

Al utilizar este enfoque, no se necesita un CDS especial y tampoco se necesitan reglas especiales para garantizar que dos sistemas HSC independientes puedan compartir recursos de hardware. No obstante, este método requiere que la prueba de DR se ejecute con datos actuales o, al menos, con datos actualmente disponibles. Tenga en cuenta que la salida de la prueba de DR y los VTV replicados por TAPEPLXA usarán espacio del buffer del VTSS de TAPEPLXB.

Dado que los datos que se replicaron a partir del TapePlex TAPEPLXA son de solo lectura, cualquier intento de la prueba de DR por modificar los datos generará el mensaje SMC0247, "Mount failed for write-protected VTV vvvvvv on drive dddd from SMC indicating that the VTV cannot be mounted" ("El montaje falló para el VTV vvvvvv protegido contra escritura en la unidad dddd de SMC, lo que indica que el VTV no se puede montar"). La aparición de este mensaje puede indicar que el proceso de DR no tiene puntos de control de la aplicación definidos claramente (consulte "[Consideraciones de VTV de CRT de solo lectura](#)"). Si este es el caso, el uso de CTR para su estrategia de DR puede no ser una buena opción.

Nota:

Si utiliza replicación cruzada entre sistemas TapePlex para crear copias de los VTV en un sitio remoto, se recomienda que no use CDRT para las pruebas de DR, ya que el uso de CDRT no permitirá que los VTV de solo lectura se actualicen, incluso en un entorno de CDRT independiente.

Realización de pruebas de DR cuando el sitio de DR no tiene LPAR

Cuando gestiona el hardware de CTR en el sitio de DR con un TapePlex que se ejecuta en el sitio de producción, hay algunas consideraciones adicionales que se deben tener en cuenta para una prueba de DR. En este ejemplo, se utiliza el TAPEPLXA como TapePlex de producción y el TAPEPLXB como el TapePlex que, normalmente, se ejecuta en el sitio de producción, pero que se ejecuta en el sitio de DR durante una prueba de DR.

1. Debe detener el TAPEPLXB del TapePlex de DR en el sitio de producción antes de realizar la prueba.

Durante la prueba de DR, el TAPEPLXB se ejecutará en el sitio de DR, en una copia del CDS de TAPEPLXB.

2. Los VTV de producción se seguirán enviando al TAPEPLXB y se reflejarán en el CDS del sitio de DR.

Durante este período, el CDS de TAPEPLXB del sitio de producción se actualiza, ya que deja de reflejar los VTV que se replican durante la prueba de DR. Las sentencias TAPEPLEX y SERVER en las LPAR de producción garantizan que la replicación de datos continúe durante la prueba de DR:

```
TAPEPLEX NAME(TAPEPLXB)
SERVER NAME(TPLXBPR) TAPEPLEX(TAPEPLXB) HOSTNAME(MVSX) PORT(999)
SERVER NAME(TPLXBDR) TAPEPLEX(TAPEPLXB) HOST(MVSXDR) PORT(1234)
```

3. Cuando inicia el HSC/VTCS para el TapePlex TAPEPLXB en el sitio de DR, debe asegurarse de iniciar el servidor HTTP en SMC:

```
HTTP START PORT(1234)
```

El número de puerto (1234) coincide con lo definido en la sentencia *TAPEPLXBDR SERVER*.

4. Al final de la prueba reutilice todos los VTV creados por la prueba.

No debe determinar los VTV que fueron creados por la prueba, simplemente, puede reutilizar todos los volúmenes de las subagrupaciones. Por ejemplo:

```
SCRATCH VOL(BV1000-BV2999)
```

5. Detenga el HSC/VTCS para TAPEPLXB en el sitio de DR.
6. Debe asegurarse de que el CDS de TAPEPLXB del sitio de DR se envíe de regreso al sitio de producción.

En una situación ideal, esto se puede hacer reflejando el CDS de TAPEPLXB nuevamente al sitio de producción durante la prueba de DR. Si esto no es posible, puede usar un FTP u otro mecanismo de su preferencia para copiar el CDS de la versión actual del sitio de DR nuevamente en el sitio de producción.

7. Reinicie el TAPEPLXB en las LPAR del sitio de producción.

Si bien no hay una copia activa del TAPEPLXB, los VTV programados para CTR a TAPEPLXB permanecerán en el buffer del VTSS. Cuando el TAPEPLXB esté activo nuevamente en el sitio de producción, estos VTV se replicarán al VTSS del sitio de DR.

Gestión de VTV replicados mediante replicación cruzada entre sistemas TapePlex (CTR)

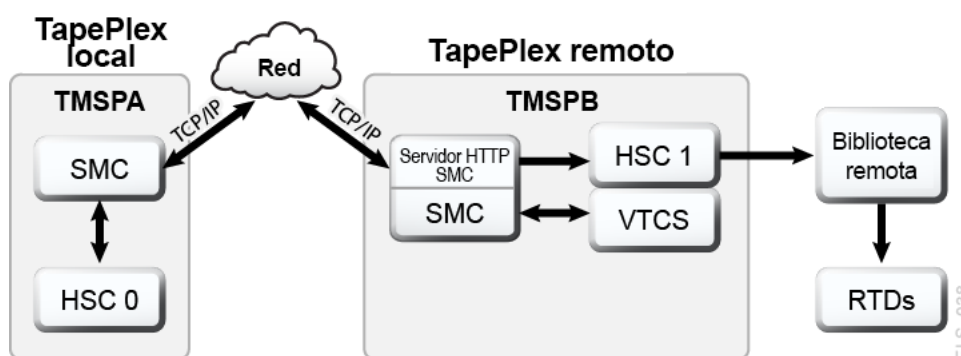
Puede usar *VTVMaint* para cambiar el estado de los VTV replicados mediante CTR de la siguiente manera:

- Use *VTVMaint DELEXPot* para eliminar el nombre de un sistema TapePlex que hace referencia a un VTV. Por ejemplo, si replica un VTV de TAPEPLXA a TAPEPLXB y, luego, suprime la copia del TAPEPLXA, puede usar *VTVMaint DELEXPot* para eliminar la referencia del TAPEPLXA al VTV.
- Use *VTVMaint ADDEXPot* para agregar el nombre de un TapePlex que hace referencia a un VTV como se describe en "[Uso de CTR para continuidad empresarial.](#)"
- Use la utilidad *VTVMaint* para cambiar la propiedad de un VTV que se recibió mediante CTR, pero el VTV debe estar actualmente en estado reutilizable. Por ejemplo, *VTVMaint OWNRPlex(TAPEPLXB)* cambiará la propiedad de un VTV enviado de TAPEPLXA que debe ser propiedad del TapePlex donde reside actualmente.

Capítulo 5. Configuración de una biblioteca remota

En [Figura 5.1, “Unidades de cinta reales \(RTD\) en funcionamiento en un TapePlex remoto”](#), se muestra un ejemplo de configuración con una biblioteca remota. En las secciones siguientes, se indica cómo configurar este ejemplo.

Figura 5.1. Unidades de cinta reales (RTD) en funcionamiento en un TapePlex remoto



Modificación del archivo SMC SCMDS

SMC gestiona toda la comunicación entre VTCS y un TapePlex remoto, por lo tanto, SMC debe saber cómo conectarse al TapePlex remoto. Para hacerlo, debe definir un TapePlex local y uno remoto, además de una o más sentencias *SERVER de1 SMC* que definen las rutas de control de TCP/IP al TapePlex remoto. Es posible que desee hacer esto en su archivo *SMC CMDS*, como se muestra en el siguiente ejemplo.

```
TAPEPLEX NAME(TMVSA)LOCSUB(HSC0)
TAPEPLEX NAME(TMVSB)LOCSUB(HSC1)
SERVER NAME(TMVSB)IP(192.168.1.10)PORT(60000)
```

Este ejemplo incluye:

- Una sentencia *TAPEPLEX*, que define un TapePlex local, TMVSA, con un HSC que se ejecuta en el mismo host de MVS (HSC0).
- Una segunda sentencia *TAPEPLEX*, que define un TapePlex remoto, TMVSB, con un HSC que se ejecuta en el host remoto de MVS (HSC1).
- Un comando *SERVER* que define una ruta de comunicación de UUI con TMVSB, donde:

- El nombre del servidor remoto es TMVSB.
- El valor del parámetro *IP* es la dirección IP del puerto de ELS de 192.168.1.10 para las comunicaciones de UUI.
- El valor del parámetro *PORT* es 60000. Este valor siempre se utiliza para el parámetro *SERVER PORT* para la comunicación SMC con un TMVSB.

Actualización de la plataforma VTCS CONFIG para definir una biblioteca remota

Debe actualizar la plataforma *VTCS CONFIG* para definir la biblioteca remota y la conectividad del VTSS a la biblioteca remota. Como se muestra en el siguiente ejemplo, la biblioteca remota se define mediante una sentencia *CONFIG STORMNGR*.

```
TAPEPLEX THISPLEX=TMVSA
STORMNGR NAME=TMVSB
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTDPATH NAME=RM1RTD1 STORMNGR=TMVSB DEVNO=1A00 CHANIF=0A:0
RTDPATH NAME=RM1RTD2 STORMNGR=TMVSB DEVNO=1A01 CHANIF=0A:1
RTDPATH NAME=RM1RTD3 STORMNGR=TMVSB DEVNO=1I00 CHANIF=0I:0
RTDPATH NAME=RM1RTD4 STORMNGR=TMVSB DEVNO=1I01 CHANIF=0I:1
VTD LOW=6900 HIGH=69FF
```

En este ejemplo, tenga en cuenta lo siguiente:

- La sentencia *CONFIG TAPEPLEX* especifica TMVSA como el TapePlex local.
- Una sentencia *STORMNGR* especifica TMVSB como la biblioteca remota.
- Las sentencias *CONFIG RTDPATH* para VTSS1, que especifican:
 - El nombre de *RTDPATH*.
 - Las conexiones con la biblioteca remota (*STORMNGR=TMVSB*).
 - El número de dispositivo (*DEVNO*).
 - El valor *CHANIF* para cada VTSS para a conexión RTD en formato *ci:p*, donde:
 - > *c* es 0 o 1.
 - > *i* es A o I.
 - > *p* es de 0 a 3.

Nota:

En los VSM5, el valor *CHANIF* debe coincidir con los valores especificados en la pantalla de estado de configuración de IFF de VSM5. En los VSM 6, este valor debe ser único para cada VTSS, pero no corresponde a un valor real de los puertos TCP/IP de VSM 6.

Ahora, puede usar el parámetro *STORcLas STORMNGR* para enrutar datos a la biblioteca remota. Por ejemplo:

```
STOR NAME(REMLIB) STORMNGR(TMVSB)
```

Consideraciones sobre la agrupación de MVC

En una biblioteca remota, los MVC se deben incluir en las definiciones de la agrupación local de MVC, preferiblemente mediante las definiciones *VOLPARM/POOLPARM*. El HSC local considera que los MVC remotos no pertenecen a la biblioteca.

Generalmente, un servidor remoto de biblioteca debe definir los MVC como "gestionados de manera externa" por algún método. No se admite que un TapePlex remoto comparta una agrupación de MVC locales, porque no hay serialización.

Si se comparte una agrupación de MVC con fines de transferencia de datos, un solo TapePlex puede escribir activamente en los MVC. Los demás TapePlex pueden tener solamente acceso de solo lectura. La serialización de los MVC en las unidades es responsabilidad del usuario.

Capítulo 6. Uso de configuraciones de VTSS en cluster

Los VTSS en cluster le permiten copiar VTV de un VTSS a otro. El VTSS en cluster es una herramienta útil para aplicaciones como, por ejemplo, soluciones de recuperación ante desastres (DR). En estas secciones, se explican los aspectos básicos de los clusters de VTSS y cómo funcionan:

- ["¿Qué es un VTSS en cluster?"](#)
- ["Requisitos de VTSS en cluster"](#)
- ["Cómo funcionan las configuraciones de VTSS en cluster"](#)

Después de la información básica, hay variaciones y funciones adicionales de los VTSS en cluster que usted puede desear conocer:

- ["Clusters unidireccionales y bidireccionales"](#)
- ["Agrupación en clusters ampliada"](#)
- ["Replicación sincrónica o asincrónica"](#)
- ["Agrupación en clusters con conexiones TCP/IP"](#)

¿Qué es un VTSS en cluster?

Un cluster de VTSS es una solución de alta disponibilidad (HA) que permite una máxima disponibilidad de los datos. Consta de dos o más sistemas VTSS conectados mediante enlaces de comunicación (CLINK) FICON o TCP/IP. Además, cada sistema VTSS dentro del cluster puede acceder a todos los datos creados dentro del cluster (VTSS residente o migrado). Los datos (VTV) creados en un cluster se replican de un sistema VTSS a otro VTSS dentro del mismo cluster bajo el control de políticas de VTCS.

Nota:

Para garantizar que todos los sistemas VTSS puedan acceder a todos los datos creados dentro del cluster, las configuraciones en cluster pueden ser cualquiera de las siguientes:

- Cada VTSS del cluster tiene RTD o VLE conectadas.
 - "Sin cintas": ningún VTSS tiene VLE o RTD conectadas.
-

Por lo tanto, las configuraciones en cluster ofrecen la más alta disponibilidad de datos con una recuperación en caliente si un VTSS dentro de un cluster sufre una interrupción (los datos replicados permanecen disponibles sin necesidad de recuperación a partir de los MVC).

Antes de VTCS 7.0, un cluster solo podía alojar dos VTSS. Con VTCS 7.0, varios VTSS pueden conformar un solo cluster. No obstante, un VTV, solo puede residir en dos VTSS en cualquier momento determinado.

Un cluster puede abarcar varias ubicaciones geográficas. Sin embargo, un cluster **debe estar dentro de un solo TapePlex** (controlado por un solo CDS).

Un VTV se puede replicar (copiar) de un VTSS a otro:

- De manera asincrónica respecto de la creación del VTV. Se programa para completarse tan pronto como sea posible después del desmontaje del VTV.
- De manera sincrónica respecto de la creación del VTV. El desmontaje del VTV no finalizará hasta que haya finalizado la replicación.

Nota:

Para la replicación sincrónica, mediante CLINK o mediante una replicación mejorada, el VTSS termina la replicación si la replicación causará que se supere el manejador de interrupciones faltante (MIH).

Para VSM 6 y VSM 7, el valor de timeout de MIH se puede configurar de 20 a 45 minutos con incrementos de 5 minutos. Para VSM4 y VSM 5, el valor de timeout de MIH se puede configurar en 20 o 45 minutos. El valor configurado en el VTSS debe coincidir con el valor configurado en el mainframe.

Las conexiones entre los VTSS dentro de un cluster pueden ser unidireccionales, donde los datos (VTV) fluyen solo en una dirección, o bidireccionales, donde los datos (VTV) puede fluir en ambas direcciones. La utilidad CONFIG especifica si un cluster es unidireccional o bidireccional, y la clase de gestión de los VTV determina su política de replicación, si corresponde, y si la replicación se realiza de manera sincrónica o asincrónica.

Por lo tanto, el almacenamiento de MVC (descrito en "[Uso de la función de almacenamiento externo de ELS](#)") y la replicación de VTV pueden facilitar una solución de recuperación ante desastres/continuidad empresarial. No obstante, la replicación de VTV es superior como solución de alta disponibilidad, porque gracias a la replicación:

- Se pueden realizar copias de seguridad de los datos de manera sincrónica.
- Los datos recientes que se han replicado en un VTSS de "recuperación", se pueden restaurar más rápidamente porque no necesita montar MVC.

Requisitos de VTSS en cluster

En [Tabla 6.1, "Requisitos de VTSS en cluster"](#), se muestran los requisitos de VTSS en cluster.

Tabla 6.1. Requisitos de VTSS en cluster

Componente	Requisito
Clusters ampliados	D02.07.00.00 o microcódigo superior del VTSS para VSM4 o VSM5 con conexiones FICON únicamente. Para VSM 6 y VSM 7, todos los niveles de microcódigo.
2 VTSS dentro de un cluster (interfaces ESCON)	Los VTSS principal y secundario pueden ser cualquier combinación de VSM4, donde el secundario puede tener cualquier capacidad. Los VSM5 no tienen interfaces ESCON y no pueden estar en un cluster con otros VTSS que usan ESCON
2 VTSS dentro de un cluster (interfaces FICON)	Los VTSS principal y secundario pueden ser cualquier combinación de VSM4 y VSM5, donde el secundario puede tener cualquier capacidad. Por ejemplo, todo lo que se indica a continuación es válido: <ul style="list-style-type: none"> • VSM5 principal, VSM4 secundario • VSM5 principal, VSM5 secundario • VSM4 principal, VSM4 secundario • VSM4 principal, VSM5 secundario (no recomendado)
Microcódigo del VTSS principal y secundario	El microcódigo del VTSS principal debe estar en un nivel que admita el envío de VTSS replicados. El microcódigo del VTSS secundario debe estar en un nivel que admita la recepción del VTSS replicado y el uso del secundario como VTSS de producción. Después de instalar el microcódigo, se debe activar la función de agrupación en clusters en ambos VTSS, el principal y el secundario, con un disquete de opciones. Consulte a su representante del servicio de hardware de StorageTek para obtener más información.
VTD reservados para agrupación en clusters	En las configuraciones de VTSS en cluster, debe asegurarse de que los primeros 16 VTD de cada VTSS (0-F) se reserven para agrupación en clusters. Estos dispositivos deben estar FUERA DE LÍNEA para los MVS, y sus rutas deben estar en línea para cada host de servidor de HSC. Esto también se aplica a los VTSS involucrados en la replicación cruzada entre sistemas TapePlex. VTCS no registra los primeros 16 VTD con SMC/HSC, lo cual impide montar los VTV en estos VTD.
RTD	En los entornos con dos ACS, los mismos tipos de dispositivos deben estar representados en las RTD conectadas a cada ACS, de modo que los datos migrados por un VTSS en un cluster puedan ser recuperados por el otro VTSS en el cluster. La cantidad de MVC, la ubicación y el tipo de medios usados para la migración se determinan mediante el parámetro <i>MIGPOL</i> de la sentencia MGMTclas. Se debe tener en cuenta cada ACS y, si un tipo de unidad en un ACS está conectada a uno de los VTSS en un entorno de VTSS en cluster, debe haber una unidad del mismo tipo y en el mismo ACS conectada a todos los demás VTSS de ese entorno en cluster.
IP nativo (agrupación en clusters con TCP/IP)	El IP nativo requiere CDSLEVEL F y versiones superiores, con los siguientes PTF: <ul style="list-style-type: none"> • Para 6.2: <ul style="list-style-type: none"> L1A00P7 - SMC6200 L1H14IM - SMS6200 L1H14O2 - SOS6200

Componente	Requisito
	L1H14IL - SWS6200 • Para 7.0, L1H150G (SES7000) • Para 7.1 y superiores, se incluye compatibilidad en el nivel base. Para IP nativo, se admiten las siguientes conexiones: • VSM5 a VSM5 • VSM5 a VSM 6 • VSM 6 a VSM 6 • VSM 6 o VSM5 a VLE

Los requisitos de la replicación sincrónica, que se aplica solamente a VSM4 y versiones superiores, son los que se describen en [Tabla 6.2, “Requisitos de replicación sincrónica”](#).

Tabla 6.2. Requisitos de replicación sincrónica

Requisito de replicación sincrónica	Nivel de microcódigo de VTSS	Nivel de CDS
IP nativo o puertos FICON para los CLINK	D02.03.00.00 o superior para VSM4 y VSM5. Para VSM 6 y VSM 7, todos los niveles de microcódigo	"F" o superior

Los requisitos de la replicación mejorada, que se aplica solamente a VSM6 y versiones superiores, son los que se describen en [Tabla 6.3, “Requisitos de la replicación mejorada”](#).

Tabla 6.3. Requisitos de la replicación mejorada

Requisito de la replicación mejorada	Nivel de microcódigo de VTSS	Nivel de CDS
IP nativo o puertos FICON para los CLINK	VSM 6.3.0.03.000 (se recomienda 6 .3.0.05.000) para VSM 6; VSM 7.0.0 .05.000 para VSM 7. Se aplica PTF L1H18MB.	"H" o superior

Cómo funcionan las configuraciones de VTSS en cluster

Puede usar VSM para conectar dos VTSS mediante enlaces de cluster (CLINK) para formar una *Configuración de VTSS en cluster*. Debe usar las mismas sentencias para implementar una configuración en cluster:

- Los clusters pueden ser unidireccionales o bidireccionales según las sentencias *CLINK*.
- El VTSS secundario (o segundo par) puede estar en la misma ubicación física que el principal (o primer par) o en una ubicación remota.
- La sentencia *CONFIG CLUSTER* especifica los VTSS que forman el cluster.
- La sentencia *CONFIG CLINK* define los CLINK que conectan los VTSS. La manera en que escribe las sentencias *CLINK* determina si una replicación es unidireccional o bidireccional.

Para obtener ejemplos, consulte "[VTSS en cluster unidireccional](#)" y "[VTSS en cluster bidireccional](#)."

- El parámetro *MGMTclas REPLICAT* identifica la clase de gestión que contiene los VTV que VSM replica (copia) de un VTSS del cluster al otro.

El parámetro *CONFIG GLOBAL REPLICat* ahora especifica cuándo replicar un VTV de la siguiente manera:

REPLICat

Especifica cuándo el VSM replica el VTV.

ALWAYS

La solicitud de replicación se agrega a la cola de replicación del VTCS cada vez que el VTV se desmonta, independientemente de que el VTV haya cambiado durante el montaje (el valor predeterminado).

CHANGED

La solicitud de replicación se agrega a la cola de replicación del VTCS si el VTV:

- se cambió durante el montaje o
- se encontraba en modo de solo lectura durante el montaje, pero existen menos copias de MVC del VTV de lo previsto.

Independientemente de la configuración de *CONFIG GLOBAL REPLICat*, la replicación **también** exige lo siguiente:

- El VTV debe estar desmontado en un VTSS que admita replicación, y no puede haber una copia idéntica del VTV en el otro VTSS del cluster.
- Además del valor *CONFIG GLOBAL REPLICat*, **debe** especificar *REPLICAT(YES)* en la clase de gestión de un VTV para que se produzca la replicación.

Para obtener más información, consulte la *Referencia de comandos, sentencia de control y utilidades de ELS*.

- VTCS migra inmediatamente (con *KEEP*) los VTV replicados. Puede especificar el VTSS de origen para la migración de los VTV replicados en el parámetro *MIGRATE* de la sentencia *STORclas*. **También debe tener en cuenta** que **debe** especificar la replicación en una clase de gestión que apunta a una clase de almacenamiento con un valor de parámetro *MIGRATE* para realizar la migración a partir del VTSS deseado. De lo contrario, es posible que no se produzca la migración a partir del VTSS deseado.

Dado que el VTCS migra inmediatamente (con *KEEP*) VTV replicados, independientemente de la configuración de *MGMTclas IMMDELAY*, StorageTek **recomienda enfáticamente** que **no** establezca explícitamente una política *MGMTclas IMMDELAY* para VTV replicados. Si lo hace, el VTCS cumple con la solicitud explícita de migración inmediata y migra inmediatamente el VTV afectado desde cualquier VTSS que pueda realizar la migración en primer lugar (es decir, el primer VTSS que tenga una

copia residente de VTV y una RTD disponible para realizar la migración). Por lo tanto, la configuración de una política *MGMTclas IMMDELAY* explícita es redundante y puede interferir en la migración y la replicación óptimas del VTV.

También debe tener en cuenta que la migración inmediata (*KEEP*) después de una replicación **no es lo mismo** que la automigración. Es decir, durante la migración inmediata implícita, no se suprimen VTV de ningún VTSS para gestionar la DBU. En cambio, los VTV, simplemente, se "prealmacenan de manera provisional" mediante la migración a un MVC a partir del VTSS receptor, y dejan **ambos** contenidos del buffer de VTSS sin modificaciones. Para gestionar el espacio en un cluster de VTSS, VTCS realiza una automigración de VTV, de acuerdo con el ciclo de migración y la gestión del espacio de **cualquier** VTSS. Si la capacidad del VTSS receptor es mayor o igual a la del VTSS emisor, mediante la automigración del VTSS emisor se suprime un VTV replicado de ambos VTSS. Si la capacidad del VTSS receptor es menor que la del VTSS emisor, la automigración puede comenzar en el VTSS receptor. En este caso, mediante la automigración se suprime un VTV replicado solo del VTSS receptor y se deja la copia para que siga siendo residente en el VTSS emisor.

- **Tenga en cuenta que** los requisitos de replicación de los datos se determinan después de un desmontaje, **no** después de una recuperación. La recuperación de un VTV no genera una replicación; por lo tanto, la recuperación bajo demanda y *MVCdrain* no generarán una replicación. No obstante, si el VTV se recupera y se monta en un VTD, en el momento del desmontaje se replicará en el VTSS secundario o par.
- Un cluster puede admitir distintas cargas de trabajo en cada uno de los cuatro modos operativos. Por ejemplo, solo los cluster de funcionamiento completo pueden admitir la replicación activa, pero en el modo primario degradado, puede cambiar el estado de los VTD secundarios a en línea para que MVS tome el control de la carga de trabajo. Puede usar *Query* para mostrar el estado del VTSS, la replicación del VTV, el enlace del cluster y el cluster. Puede usar *VARY VTSS* para cambiar los estados de VTSS y *VARY CLink* para cambiar los estados de CLINK.

Cómo funciona la conciliación de VTSS

- Siempre que un par de VTSS en cluster reanuda el estado de funcionamiento completo, VTCS concilia el contenido de los dos VTSS. Esto ocurre durante la inicialización del VTCS o cuando un VTSS se coloca en línea y su par también.
- La conciliación consiste en suprimir, o migrar y suprimir VTV (o replicar un VTV si la replicación no se completó anteriormente de forma correcta). Es decir, que no hay recuperación involucrada en la conciliación del contenido de VTSS.

Por ejemplo, en un cluster unidireccional con un VTV residente en el VTSS receptor, pero no en el emisor, VTCS suprime el VTV del receptor (después de asegurarse de que todas las copias necesarias del MVC se hayan hecho). De esta manera, se evita la recuperación en el emisor.

De manera similar, en un cluster unidireccional con un VTV residente en el emisor pero no en el VTSS receptor, VTCS replica el VTV en el receptor, en lugar de recuperarlo a partir de un MVC.

- El proceso de conciliación supone que si un VTV replicado reside en el VTSS emisor, entonces es una copia válida. Si la copia que se encuentra en el receptor es distinta, el VTCS la suprime.
- Para mantener acciones de conciliación coherentes en un cluster bidireccional, el VTSS en el que el VTV residía o en el que residió por última vez (según lo indicado por el registro de VTV del CDS), se considera el VTSS emisor. El proceso de conciliación es el que se describe anteriormente para los clusters unidireccionales.

Clusters unidireccionales y bidireccionales

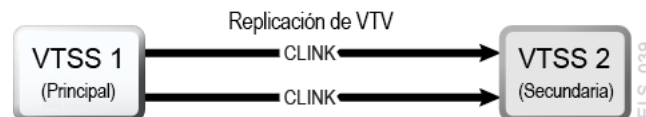
Los clusters de los VTSS pueden ser:

- Unidireccionales, donde un VTSS es el principal y otro, el secundario. Para obtener más información, consulte "[VTSS en cluster unidireccional](#)."
- Bidireccionales, donde ambos VTSS son pares y la replicación se realiza de uno a otro en cualquier dirección. Para obtener más información, consulte "[VTSS en cluster bidireccional](#)."

Clusters unidireccionales

Como se muestra en [Figura 6.1, "VTSS en cluster unidireccional"](#), en un cluster unidireccional, la replicación solo se realiza del principal al secundario.

Figura 6.1. VTSS en cluster unidireccional



Cómo funcionan los clusters de VTSS unidireccionales

- El secundario puede recibir ambos VTV replicados del principal y carga de trabajo de producción no replicada mediante ninguno de los métodos de enrutamiento estándares (por ejemplo, *TAPEREQs*). Debe cambiar el estado de los VTD en el secundario a en línea para MVS, de modo que el secundario pueda aceptar trabajo de producción. No puede cambiar el estado de las direcciones de VTD utilizadas por las terminaciones de CLINK a en línea para MVS, como se describe en "[Cómo funcionan las configuraciones de VTSS en cluster](#)."
- Un VTV con replicación activada se asigna a un VTSS principal en línea, a menos que no haya ninguno disponible; en ese caso, el VTV se asigna a un VTSS secundario en línea. Si no hay VTSS secundarios disponibles, el VTV se asigna a un VTSS sin cluster. Un VTV

sin replicación se puede asignar a cualquier VTSS en línea, incluido el secundario de un cluster de funcionamiento completo.

- En el momento del desmontaje, un VTV con replicación activada que reside en un cluster de funcionamiento completo se agrega a la cola para replicación en el VTSS secundario. Si un VTV con replicación activada se desmonta de un VTD en un VTSS que no es parte de un cluster de funcionamiento completo, el VTV se agrega a la cola para migración inmediata.

Cuando el VTSS secundario recibe un VTV replicado de un VTSS principal, el VTV se migra de inmediato (con la opción *KEEP*) independientemente de la configuración de migración inmediata de la clase de gestión para este VTV.

- **Tanto el VTSS principal como el secundario** pueden gestionar todas las recuperaciones de espacio.
- Si está utilizando interfaces ESCON o FICON, en el VTSS principal, los CIP/FIP de CLINK están configurados en *modo Nearlink*, mientras que en el VTSS secundario, los CIP/FIP están configurados en *modo de host*.

Por lo tanto, debe configurar *CLINK* solo para el VTSS principal, como se muestra en el siguiente ejemplo, donde VTSS1 es el VTSS principal.

```

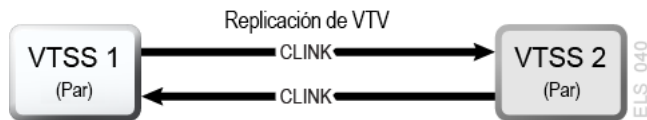
.
.
.
CLUSTER NAME=CLUSTER1 VTSSs(VTSS1,VTSS2)
CLINK VTSS=VTSS1 CHANIF=0G
CLINK VTSS=VTSS1 CHANIF=00
CLINK VTSS=VTSS1 CHANIF=1G
CLINK VTSS=VTSS1 CHANIF=10
.
.

```

Clusters bidireccionales

Como se muestra en [Figura 6.2, “VTSS en cluster bidireccional”](#), la agrupación en clusters bidireccionales requiere pares de *CLINK* unidireccionales, de modo que los datos fluyan en direcciones opuestas en los *CLINK*.

Figura 6.2. VTSS en cluster bidireccional



Cómo funcionan los clusters de VTSS bidireccionales

Durante el funcionamiento normal de un cluster bidireccional, ambos VTSS están en línea para el VTCS, como se muestra a continuación:

- En un cluster bidireccional, cada uno de los VTSS pares puede recibir trabajo de producción mediante métodos de enrutamiento estándares (por ejemplo, TAPEREQs). Debe cambiar el estado de los VTD en ambos VTSS a en línea para MVS, de modo que cada uno pueda aceptar trabajo de producción. No obstante, **tenga en cuenta que no** puede cambiar el estado de las direcciones de VTD utilizadas por las conexiones de CLINK a en línea, como se describe en " [Cómo funcionan las configuraciones de VTSS en cluster.](#)"
- En un cluster bidireccional, un VTV con replicación activada se asigna a cualquiera de los VTSS pares. Si uno de los VTSS pares está fuera de línea o desactivado, la carga de trabajo de producción se puede ejecutar en el resto de los VTSS que estén en línea. No obstante, los VTV que requieren replicación, se asignan a los VTSS restantes solamente si no hay otros clusters de funcionamiento completo disponibles o adecuados. En este caso, los VTV replicados se migran de inmediato mediante la función KEEP y se los agrega a la cola para replicación hasta el momento en que el VTSS esté en línea.
- En un cluster bidireccional, en el momento del desmontaje, un VTV con replicación activada que reside en un cluster de funcionamiento completo se agrega a la cola para replicación en el otro VTSS par. Si un VTV con replicación activada se desmonta de un VTD en un VTSS que no es parte de un cluster de funcionamiento completo, el VTV se agrega a la cola para migración inmediata. **Tenga en cuenta que** los requisitos de replicación de los datos se determinan después de un desmontaje, **no** después de una recuperación. La recuperación de un VTV no genera una replicación; por lo tanto, la recuperación bajo demanda y MVCdrain no generarán una replicación. No obstante, si el VTV se recupera y se monta en un VTD, en el momento del desmontaje se replicará en el VTSS secundario, a menos que especifique REPLICAT(CHANGED) (la opción recomendada), como resultado de lo cual el VTV se replicará nuevamente solo si los datos cambiaron.
- Los dos VTSS pares pueden gestionar recuperaciones de espacio.
- Si está usando interfaces ESCON o FICON:
 - En cada VTSS par, los *CLINK CIPs/FIPs* "emisores" están configurados en *modo Nearlink*, mientras que los *CLINK CIPs/FIPs* "receptores" están configurados en *modo de host*.

Por lo tanto, debe configurar los *CLINK* "emisores" en cada VTSS par, como se muestra en el siguiente ejemplo, donde VSMR1 y VSMR2 son VTSS pares.

```

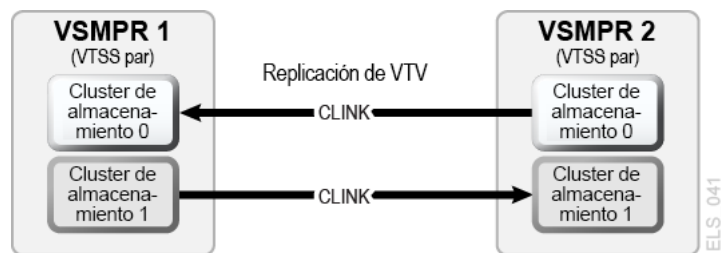
.
.
CLUSTER NAME=CLUSTER1 VTSSs(VSMR1,VSMR2)
  CLINK VTSS=VSMR1 CHANIF=00:0
  CLINK VTSS=VSMR1 CHANIF=00:1
  CLINK VTSS=VSMR2 CHANIF=10:0
  CLINK VTSS=VSMR2 CHANIF=10:1
.
.

```

- Cada CLINK debe estar conectado al *mismo cluster de almacenamiento* de cada VTSS (cluster de almacenamiento 0 a cluster de almacenamiento 0 o cluster de almacenamiento 1 a cluster de almacenamiento 1). Si no puede realizar la configuración de esta manera, se pueden producir errores de replicación, canal y comunicación.

Como se muestra en el ejemplo de [Figura 6.3, “CLINK ESCON/FICON para VTSS en cluster bidireccionales”](#), el puerto del CLINK emisor (modo Nearlink) en VSMPR1 está en el cluster de almacenamiento 1 y se conecta a un puerto del CLINK receptor (modo de host), *que también está en el cluster de almacenamiento 1*, en VSMPR2. De manera similar, un puerto del CLINK emisor en el cluster de almacenamiento 0 de VSMPR2 se conecta a un puerto del CLINK receptor en el cluster de almacenamiento 0 de VSMPR1.

Figura 6.3. CLINK ESCON/FICON para VTSS en cluster bidireccionales

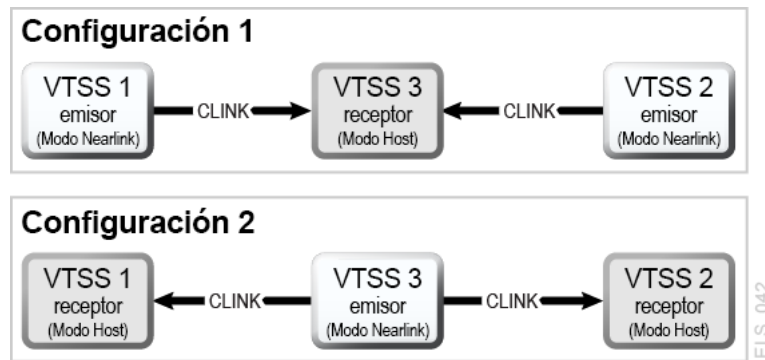


Agrupación en clusters ampliada

La agrupación en clusters ampliada (EC) permite la conexión de tres o más VTSS mediante CLINK con una sola configuración de TapePlex (1 CDS). La agrupación en clusters es una solución de alta disponibilidad diseñada para que una carga de trabajo pueda continuar sin interrupciones durante una interrupción del VTSS. La agrupación en clusters exige que todos los subsistemas VTSS que formen parte de un cluster tengan acceso a todos los MVC generados por un solo subsistema VTSS en ese cluster. Si un VTSS dentro de un cluster se conecta a un Tapeplex remoto (CTR), todos los subsistemas VTSS del cluster se deben conectar al mismo TapePlex para conservar la capacidad de HA.

Mediante la agrupación en clusters ampliada, puede configurar un VSM con CLINK conectados a varios VTSS y la cantidad de conexiones de CLINK solo se ve limitada por la cantidad de conexiones físicas disponibles. Se requiere D02.07.00.00 o microcódigo superior. Todas las reglas de agrupación en clusters y replicación disponibles se aplican a EC. Todas las configuraciones de una agrupación en clusters ampliada se establecen en función de dos configuraciones básicas unidireccionales, como se muestra en [Figura 6.4, “Configuraciones básicas de una agrupación en clusters ampliada”](#).

Figura 6.4. Configuraciones básicas de una agrupación en clusters ampliada



Replicación sincrónica o asincrónica

Puede elegir entre una de las siguientes opciones: puede replicar de manera sincrónica o asincrónica, en función de las políticas de su sitio.

Implementación de la replicación sincrónica

Atención:

Con la replicación sincrónica, el tiempo necesario para replicar un volumen virtual demorará la finalización de cualquier trabajo que genere datos que tengan una política de replicación sincrónica.

1. Asegúrese de que el sistema cumpla con los requisitos de replicación sincrónica que se describen en [Tabla 6.2, “Requisitos de replicación sincrónica”](#).
2. Con todos los sistemas HSC/VTCS apagados, use `CONFIG GLOBAL` para activar la replicación sincrónica:

```
CONFIG GLOBAL SYNCHREP=YES
```

3. Asegúrese de que el parámetro `CONFIG GLOBAL REPLICAT` esté establecido como lo desea:

ALWAYS

La solicitud de replicación se agrega a la cola de replicación del VTCS cada vez que el VTV se desmonta, independientemente de que el VTV haya cambiado durante el montaje (el valor predeterminado).

CHANGED

La solicitud de replicación se agrega a la cola de replicación del VTCS si el VTV:

- se cambió durante el montaje o
- se encontraba en modo de solo lectura durante el montaje, pero existen menos copias de MVC del VTV de lo previsto.

4. Especifique la replicación sincrónica en las sentencias `MGMTCLAS` que desee:

```
MGMT (name) . . . . . REP(YES_SYNC)
```

Implementación de replicación asincrónica con supervisión de trabajos

Puede optar por usar la replicación asincrónica, pero también es posible que quiera saber si la replicación finalizó correctamente. En este procedimiento, debe usar la utilidad *DRMONitr* para pausar el trabajo de MVS asociado hasta que la replicación finalice correctamente.

1. Asegúrese de que el sistema cumpla con los requisitos de replicación sincrónica que se describen en [Tabla 6.2, “Requisitos de replicación sincrónica”](#).
2. Con todos los sistemas HSC/VTCS apagados, use *CONFIG GLOBAL* para activar la replicación asincrónica:

```
CONFIG GLOBAL SYNCHREP=NO
```

3. Asegúrese de que el parámetro *CONFIG GLOBAL REPLICAT* esté establecido como lo desea:

ALWAYS

La solicitud de replicación se agrega a la cola de replicación del VTCS cada vez que el VTV se desmonta, independientemente de que el VTV haya cambiado durante el montaje (el valor predeterminado).

CHANGED

La solicitud de replicación se agrega a la cola de replicación del VTCS si el VTV:

- se cambió durante el montaje o
- se encontraba en modo de solo lectura durante el montaje, pero existen menos copias de MVC del VTV de lo previsto.

4. Especifique la replicación asincrónica en las sentencias *MGMTCLas* que desee:

```
MGMT (mgmtname) ..... REP(YES)
```

5. Cree un JCL para supervisar la replicación asincrónica.

Para esto, use la utilidad *DRMONitr* para supervisar la replicación. *DRMONitr* hace que el trabajo de MVS asociado pause hasta que la replicación finalice correctamente. Por ejemplo:

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
/** If HSC IS NOT OR MAY NOT BE ACTIVE, INCLUDE THE
/** FOLLOWING:
//SLSCNTL DD DSN=primary.cds.name, DISP=SHR
//SLSCNTL2 DD DSN=secondary.cds.name, DISP=SHR
//SLSSTBY DD DSN=standby.cds.name, DISP=SHR
//SLSPARMP DD DSN=h1q.PARMLIB(BKPCNTL), DISP=SHR
//SLSPARMS DD DSN=h1q.PARMLIB(BKPCNTL2), DISP=SHR
```



```
//SLSPARMB DD DSN=h1q.PARMLIB(BKPSTBY),DISP=SHR
//SYSIN      DD UNIT=SYSDA,SPACE=(TRK,1)
//* THE FOLLOWING IS USED BY THE SNAPSHOT UTILITY:
//SYSPRINT  DD SYSOUT=*
//SLSPRINT  DD SYSOUT=*
//SLSIN     DD *
DRMON MGMT(mgmtname) REPL MAXAGE(24) TIMEOUT(120)
```

En este ejemplo, la utilidad *DRMON* supervisa las replicaciones de la clase de gestión especificada. Asimismo, supervise únicamente los VTV que se actualizaron en las últimas 24 horas y genere un timeout para *DRMON* después de 120 minutos.

Agrupación en clusters con conexiones TCP/IP

La función de conexión de IP nativo del VTSS le permite usar el protocolo TCP/IP para "agrupar en clusters" (conectar) dos o más VTSS para replicación de VTV. Con la agrupación en clusters de IP nativo, cada VTSS tiene puertos de Ethernet para conexión a la red TCP/IP. Antes, estaba limitado a las conexiones ESCON o FICON para replicación. Mediante el uso de TCP/IP para CLINK puede ofrecer un mejor rendimiento de la replicación en protocolos ESCON o FICON y, si lo desea, los puertos ESCON o FICON existentes se pueden usar exclusivamente para RTD y conexiones host, donde se admiten las siguientes conexiones:

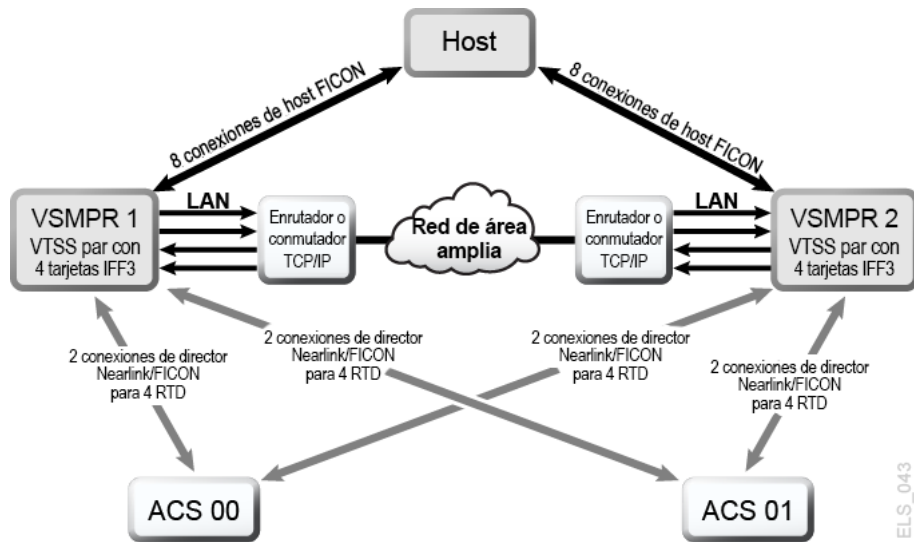
- VSM5 a VSM5
- VSM5 a VSM 6
- VSM 6 a VSM 6

En esta sección, se describe solo la implementación de VTCS para IP nativo. El personal de soporte del hardware de StorageTek u otros QSP son responsables de la configuración del VTSS.

El entorno de TCP/IP

Los CLINK conectados a TCP/IP cumplen la misma función que los CLINK de conexión de canal FICON o ESCON, pero el CLINK TCP/IP se conecta mediante un puerto Ethernet en el VTSS, en lugar de hacerlo desde un puerto ESCON o FICON. En el ejemplo que aparecen en [Figura 6.5, "Entorno de TCP/IP con dos VSM5"](#), se muestran VSM5 pares, cada uno con 4 tarjetas IFF3 con puertos Ethernet. Los cables Ethernet de los puertos Ethernet de las tarjetas IFF3 se conectan a las redes de área local (LAN, una para cada VTSS) y las LAN se conectan mediante una red de área amplia (WAN).

Figura 6.5. Entorno de TCP/IP con dos VSM5



Configuración de VTCS para CLINK TCP/IP

A continuación, se muestran los parámetros de la sentencia *CONFIG CLINK*.

Sentencia CONFIG CLINK

La sentencia *CONFIG CLINK* suministra dos tipos de conexiones de VTSS a VTSS mediante los siguientes parámetros:

CLINK CHANIF=*nn* o *nn:n*

Define un puerto FICON o ESCON para usar como CLINK.

CLINK IPIF=*ci:p*

Define un puerto Ethernet para usar como CLINK. Los valores válidos para *CONFIG RTD IPIF* son *c=0* o *1*, *i=A* o *I*, *p=0* a *3* para VSM5 y VSM 6. En los VSM5, este valor debe coincidir con los valores especificados en la pantalla de estado de configuración de IFF de VSM5. En los VSM 6, este valor debe ser único para cada VTSS; y no corresponde a un valor real de los puertos TCP/IP de VSM 6.

Nota:

La sentencia *CLINK* debe contener el parámetro *CHANIF* o el parámetro *IPIF*, pero no ambos.

Capítulo 7. Uso del software de prueba concurrente de recuperación ante desastres

El posible que los clientes que usan o mantienen un sitio de recuperación ante desastres (DR) como parte de un plan de continuidad empresarial deseen validar periódicamente su capacidad para continuar el procesamiento normal de la producción antes de que ocurra un desastre real; otros clientes no tienen opción y deben probar de forma periódica la preparación de su modelo de continuidad empresarial para cumplir con los requisitos de seguro y/o de los auditores.

Mediante el uso de la función de prueba concurrente de recuperación ante desastres (CDRT), que ahora es una función integrada al software de StorageTek ELS, las empresas que actualmente usan las bibliotecas de cintas StorageTek Streamline y/o Nearline (hardware real), VSM (hardware virtual) y software asociado (HSC, VTCS) pueden validar su capacidad de continuidad empresarial de cintas reales y virtuales sin necesidad de comprar hardware o software adicional.

CDRT admite una prueba paralela de aplicaciones y hosts de producción con acceso simultáneo a los datos de producción, mediante los sistemas de prueba de DR y producción.

Los conceptos clave de CDRT son:

- Mediante el uso de CDRT, una prueba de DR puede ejecutarse con hardware real, virtual o ambos.
- CDRT, HSC y VTCS aplican de forma programática algunas restricciones funcionales durante la preparación de CDS y la prueba real de DR en un intento por garantizar la integridad del sistema.
- CDRT separa de forma lógica una porción de hardware real y virtual y agrupaciones de volúmenes de cintas de producción existentes para el período de la prueba de DR. Esto permite probar la configuración de DR y, al mismo tiempo, ejecutar el trabajo de producción, además, permite garantizar la integridad de los datos de producción y minimizar los conflictos para los volúmenes de cinta y los recursos de hardware.
- La CDRT crea una copia de prueba del CDS de producción. Por lo tanto, el subsistema de producción de ELS y los subsistemas de prueba de DR de ELS no se comunican entre sí. Los cambios que se producen en el CDS de prueba de DR no se reflejan en la copia del CDS de producción y viceversa. Los hosts de prueba de DR ejecutan el hardware separado de forma lógica únicamente. Los hosts de producción siguen usando todo el

hardware con una excepción: los hosts de DR usan exclusivamente los VTSS separados de forma lógica durante la prueba de DR. Otros recursos, como las RTD, los cartuchos de varios volúmenes (MVC) y las cintas reutilizables reales se deben controlar mediante la definición de agrupaciones separadas para cada conjunto de hosts.

- Una prueba de DR se puede realizar solo con recursos locales o con una combinación de recursos locales y remotos; también se admiten las configuraciones que constan de un sitio remoto con hardware real y virtual solamente, o que constan de hardware real y virtual con un procesador de mainframe.
- El hardware de prueba de DR puede incluir cualquier combinación de ACS, VTSS y VLE.
- Después de una prueba de DR, la copia de prueba de CDS y todos los datos creados a partir de la prueba de DR, por lo general, se desechan, y el hardware separado lógicamente se vuelve a desplegar en el entorno de producción normal.

Nota:

- Para satisfacer las necesidades de recuperación de un verdadero desastre, es fundamental que los trabajos en un flujo de trabajos de prueba de DR **no** actualicen los volúmenes creados en la producción, ya sea mediante *DISP=MOD* o mediante la sobrescritura de estos volúmenes. El uso de dichas prácticas significa que si ocurrió un verdadero desastre, el estado de estos volúmenes será impredecible.
 - Para la ejecución de la prueba de DR, se **recomienda enfáticamente** que los volúmenes de producción que se pudieron modificar durante la prueba de DR se copien a nuevos volúmenes al comienzo de la prueba y que los volúmenes **copiados** sean actualizados por la prueba de DR, en lugar de los volúmenes originales. Además, el JCL se debe modificar, de ser posible, de modo que se conozca el estado de todos los volúmenes de cinta en el momento de un desastre.
-

Consideraciones sobre metadatos

Un factor fundamental para realizar una prueba de DR exitosa mediante CDRT es contar con una copia coherente del estado de todos los volúmenes de cinta gestionados por el software de ELS y el hardware real y virtual. La coherencia en el estado de los volúmenes de cinta entre los hosts de producción y los hosts de DR al comienzo de la prueba de DR es lo que permite el procesamiento paralelo de las aplicaciones del cliente. Dado que el CDS refleja el estado de todos los recursos y volúmenes de cinta en el hardware real y virtual, la CDRT cumple parcialmente con este requisito de coherencia cuando hace la copia de prueba del CDS.

No obstante, en un entorno de volúmenes de cintas, muchas veces, algunos de los datos del estado de estos volúmenes de cintas (metadatos) se conservan y se gestionan fuera del subsistema de ELS y el hardware real y virtual. Por lo general, los metadatos de los volúmenes de cintas (VOLSER, DSN, fecha de caducidad, estado reutilizable, designación real o virtual, etc.) se almacenan en uno o más catálogos de gestión de cintas (TMC), en uno o más catálogos de z/OS y en el CDS.

Debe coordinar la creación de copias de metadatos conservados y gestionados fuera del ELS (y el hardware real y virtual) con la creación de la copia de prueba del CDS que realiza la CDRT.

¿Dónde obtiene la CDRT los datos del VTV?

La CDRT obtiene los datos de VTV de uno o más de los siguientes recursos del sitio de DR:

- MVC
- VLE
- VTSS

Debido a que el CDS de producción es la fuente de información de los VTV disponibles para la prueba de DR, es importante asegurarse de que el ciclo de sincronización de volúmenes reutilizables permita que los volúmenes utilizados en la prueba de DR no se coloquen en el estado reutilizable antes del comienzo de la prueba. StorageTek también recomienda que cambie el VTSS de prueba a fuera de línea antes de realizar la *DRTEST CREATE*.

Tenga en cuenta que la ejecución de la reutilización durante la prueba **no** afecta al contenido del VTSS de prueba de DR ni los de los MVC, si este se establece en *READONLY* mediante la utilidad *ACTMVCGN*.

La copia de CDS del sitio de DR proporciona la ubicación de los VTV en el momento de la copia, y la ubicación, por lo general, se encuentra en los MVC o las VLE. No obstante, en ocasiones, puede optar por usar copias de VTV en un VTSS que existe en el sitio de prueba. En general, puede hacer esto en las siguientes situaciones:

- Define su VTSS de DR con la palabra clave *SHARE*, que impide actualizaciones del contenido de cualquier VTV de producción.
- Tiene dos VTSS en cluster: uno en el sitio de producción y uno en el sitio de DR; y su prueba de DR **no modifica** el contenido de ningún VTV de producción.
- Tiene dos VTSS en cluster: uno en el sitio de producción y uno en el sitio de DR; y desea dedicar tiempo, después de la prueba de DR, a identificar y migrar manualmente cualquier VTV que actualizó la prueba de DR.

Nota:

Puede usar la utilidad *DRMONitr* para garantizar que los datos críticos de DR lleguen a la ubicación de recuperación designada antes de ejecutar una prueba de DR.

¿Qué ocurre con los datos cuando finaliza la prueba?

Los datos creados por la prueba de DR se reflejan únicamente en el CDS de prueba de DR (que se desecha después de la prueba) y en los MVC de prueba de DR, que se encuentran en un rango separado de volúmenes de los MVC de producción. Además, los VTV creados por la prueba de DR permanecen en el VTSS de prueba de DR después de la prueba, a menos que realice una de las siguientes acciones:

1. Use la utilidad *SLUADMIN SCRATCH* en el entorno de prueba de DR para reutilizar (con *MGMTCLAS DELSCR(YES)*) todos los VTV en el rango de la subagrupación de prueba de DR. Esta opción requiere el uso de la característica *POOLPARM/VOLPARM*.

2. Asegúrese de que todos los VTV de producción que fueron modificados por la prueba de DR se migren a los MVC de prueba de DR mediante el comando *MIGRATE* con *DELETE(YES)* a partir del sistema de prueba de DR. Si no puede hacer esto, el sistema de producción recoge los datos que fueron modificados por la prueba de DR.
3. Solicítele al CSE de StorageTek CSE o a otro QSP que "limpie" el VTSS de prueba de DR después de la prueba para eliminar todos los datos del VTSS.

Nota:

Si elige la opción 2 o la opción 3, el contenido del VTSS **no coincidirá** con el sistema de producción, porque los VTV estarán ausentes del CDS de prueba de DR cuando este se devuelva a la producción. Si bien es el software el que controla esta situación, el rendimiento de su producción puede degradarse.

Gestión de los recursos de CDRT

En las siguientes secciones, se indica cómo gestionar los recursos de CDRT.

Recursos de volúmenes

El primer paso para gestionar los recursos de CDRT consiste en definir los volúmenes de su sistema mediante la utilidad *POOLPARM/VOLPARM*. La función simplifica la gestión general de sus volúmenes y agrupaciones de cintas, y proporciona un método de segregación de volúmenes que se escriben en una configuración de CDRT. El uso de *POOLPARM/VOLPARM* es necesario cuando se comparten los VTSS de prueba de DR y se **recomienda enfáticamente** para otros escenarios.

Nota:

La utilidad *SLUADMIN SET VOLPARM* se debe ejecutar mediante el CDS de producción y debe definir las agrupaciones de producción y prueba de DR. La utilidad *SET VOLPARM* **no** es válida para un CDS de prueba de DR.

Subagrupaciones reutilizables

Las subagrupaciones reutilizables son aplicables a todos los escenarios de prueba de DR. La siguiente sintaxis muestra el uso de las definiciones *POOLPARM/VOLPARM* para definir las subagrupaciones reutilizables de producción y prueba de DR. Al usar los mismos nombres para las subagrupaciones de producción y prueba de DR (con distintos rangos de volumen), no es necesario cambiar las políticas de producción cuando ejecuta la prueba de DR, como se muestra en el siguiente ejemplo.

```
*   SCRATCH POOLS
POOLPARM NAME(SCR1) TYPE(SCRATCH)
VOLPARM VOLSER(T11000-T11999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(SCR1) TYPE(SCRATCH) DRTEST
VOLPARM VOLSER(T12000-T12999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(SCRVT1) TYPE(SCRATCH)
```

```
VOLPARM VOLSER(V1000-V1999) MEDIA(VIRTUAL)
POOLPARM NAME(SCRVTV1) TYPE(SCRATCH) DRTEST
VOLPARM VOLSER(V2000-V2999) MEDIA(VIRTUAL)
```

Cuando define subagrupaciones reutilizables mediante la utilidad *POOLPARM/VOLPARM*, puede usar la utilidad *SLUADMIN SCRATCH* de HSC para reutilizar volúmenes de salida de prueba de DR dentro del entorno de prueba de DR. Junto con una clase de gestión que especifica **DELSCR(YES)**, la ejecución de la utilidad de reutilización elimina los VTV creados por la prueba de DR a partir del VTSS.

Recursos de MVC

Los recursos de MVC se utilizan para todos los escenarios de prueba de DR, excepto para cinta real únicamente y VSM sin cinta. En el siguiente ejemplo, se muestra el uso de las definiciones *POOLPARM/VOLPARM* para definir las agrupaciones de MVC de producción y prueba de DR.

```
* MVC POOLS
POOLPARM NAME(MVCP1) TYPE(MVC) MCVFREE(40) MAXMVC(4) THRESH(60) +
START(70)
VOLPARM VOLSER(T14000-T14999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(MVCP1) TYPE(MVC) MCVFREE(40) MAXMVC(4) THRESH(60) +
START(70) DRTEST
VOLPARM VOLSER(T13000-T13999) MEDIA(T10000T1) RECTECH(T1AE)
```

Conserve el contenido de los MVC de producción para utilizarlo en un escenario de prueba de DR con lo siguiente:

- Use la utilidad *ACTMVCGN* para definir los MVC que se utilizarán como entrada para la prueba de DR en *READONLY*, como se muestra en el siguiente ejemplo.

```
//ACTMVCG1 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: MVCMAINT READONLY(ON) STATEMENTS
//SLUSMVON DD DSN=h1q.SLUSMVON, DISP=(NEW, CATLG, DELETE),
// SPACE=(CYL, 1)
/* NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUSMVOF DD DSN=h1q.SLUSMVOF, DISP=(NEW, CATLG, DELETE),
SPACE=(CYL, 1)
/* NOTE: THE FOLLOWING STEP SELECTS ALL "ACTIVE" MVCS
/* IN ACS 01.
//SLSIN DD *
ACTMVCGN ACS(01)
/*
//ACTMVCG2 EXEC PGM=SLUADMIN, PARM='MIXED'
```

```
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
/** NOTE: EXEC MVCMAINT TO SET READONLY(ON)
```

- Mediante *VTCS CONFIG*, asegúrese de que no se ejecute la recuperación en todos los hosts de producción:

```
CONFIG HOST NAME(host) NORECLAM
```

Recursos de VTSS

Hay dos tipos de recursos de VTSS: no compartidos y compartidos.

Recursos de VTSS no compartidos

En un entorno donde los VTV se migran a los MVC o a la VLE, los recursos de VTSS se separan durante una prueba de DR. El sistema de prueba de DR tiene acceso solamente a los VTSS definidos mediante las utilidades *DRTEST PRIMEPRD/CREATE*, y estos VTSS deben permanecer fuera de línea para la producción. El comando *DRTEST START* se rechaza si un VTSS de DR está en línea en el entorno de producción.

En algunos casos, puede haber dos VTSS con el mismo nombre, tanto en el sitio de prueba como en el de producción. En este caso, el parámetro *SPARE* de la utilidad *DRTEST PRIMEPRD/CREATE* especifica que el VTSS de prueba de DR tiene el mismo nombre que un VTSS de producción, pero no es físicamente el mismo dispositivo, lo que permite que el dispositivo de producción permanezca en línea durante la prueba. Es importante que tenga en cuenta que no debe utilizar el parámetro *SPARE* para ningún otro escenario.

VTSS en cluster en CDRT

En "[Escenario 4: VTSS en cluster con sitios de producción y prueba de DR](#)", se utiliza VTSS en cluster para enviar datos al VTSS del sitio de DR. En este escenario, el cluster **no** funciona durante la prueba de DR, porque el VTSS de prueba de DR se encuentra fuera de línea para la producción.

Si los VTV de producción se modifican de alguna manera durante la prueba de DR, **debe** asegurarse de que todos los VTV modificados se eliminen del VTSS antes de volver a colocar el VTSS de prueba nuevamente en línea para producción. De lo contrario, el sitio de producción puede recoger los VTV modificados. Para evitar esta situación, **StorageTek recomienda enfáticamente** que la prueba de DR se diseñe de manera tal que los VTV de producción no sean alterados por la prueba.

En un escenario de VTSS en cluster puede no haber VTD en el VTSS de prueba accesible para el host de producción. Para permitir la comunicación mediante ECAM de la producción al VTSS de prueba de DR, especifique lo siguiente en *VTCS CONFIG*:


```
VTD LOW=XXXX HIGH=XXXX NOVERIFY
```

Preparación de un cluster de VTSS para una prueba de DR:

1. Cambie el estado del VTSS de prueba de DR a un estado inactivo. Por ejemplo:

```
VARY VTSS1 QUIESCED
```

El objetivo aquí es cerrar (de forma controlada) la replicación para VTSS1, de modo que pueda usarlo exclusivamente para la prueba de DR.

2. Supervise la replicación hasta que finalice con *Display REPLICat*. Aquí, la replicación todavía está activa:

```
VTSS  HOST  QDEPTHVTSS0  PRODUCTION    1
```

Usted sabrá que la replicación ha finalizado cuando vea lo siguiente:

```
VTSS  HOST  QDEPTH
VTSS0  PRODUCTION  0
```

3. Realice una comprobación cruzada para verificar que la replicación haya finalizado comprobando el estado del CLINK con *Display CLINK (Visualizar CLINK)*. Aquí, el CLINK todavía está activo:

```
VTSS  CLINK  STATUS USAGE      HOST
VTSS0  7    ONLINE REPLICATING PRODUCTION
VTSS0  8    ONLINE REPLICATING PRODUCTION
You know the CLINK is no longer active when you see this:
VTSS  CLINK  STATUS USAGE      HOST
VTSS0  7    ONLINE FREE
VTSS0  7    ONLINE FREE
```

4. Cambie el estado del VTSS de prueba de DR a fuera de línea:

```
VARY VTSS1 OFFLINE
```

Gestión del contenido de VTSS antes y después de la prueba de DR

Antes de comenzar una prueba de DR, **debe**:

1. Planificar la prueba de modo que la salida de la prueba no entre accidentalmente en la producción.

2. Preparar la prueba de modo que el CDS de prueba de DR coincida con el contenido del VTSS de DR.

Además, **StorageTek recomienda enfáticamente** que todos los VTSS del sitio de prueba se limpien tan pronto como sea posible después de la prueba. Esta práctica garantiza que los VTSS del sitio de prueba estén vacíos al comienzo de la próxima prueba y que ningún dato de prueba de DR permanezca en los VTSS que se devuelven a la producción.

Para limpiar los VTSS de prueba, realice alguna de las siguientes acciones:

1. No permita que la prueba de DR realice ninguna actualización (sobrescritura o agregación) en los VTV de producción y utilice *POOLPARM/VOLPARM* para definir las subagrupaciones reutilizables de prueba de DR. Mediante este método, puede ejecutar la utilidad de reutilización para reutilizar todos los VTV del rango de prueba de DR, y estos se suprimen automáticamente del VTSS (si la clase de gestión específica *DELSCR(YES)*).
2. Si la prueba de DR puede actualizar los VTV de producción, debe asegurarse de que cualquier VTSS que se utilice para la salida de prueba de DR se vacíe antes de que el VTSS regrese a la producción. Para hacer esto, debe migrar a cero en el entorno de prueba de DR o debe solicitarle a un CSE de StorageTek o a otro QSP que "limpie" el VTSS.

Cuando ejecuta la utilidad *DRTEST CREATE* para crear un CDS de prueba de DR, los metadatos del contenido de VTSS se propagan al entorno de prueba. Los VTV del VTSS de prueba de DR están disponibles para el entorno de prueba de DR.

Si el VTSS de prueba de DR se define como de reserva, el contenido del VTSS físico de DR no coincidirá con los metadatos de CDS, que hacen referencia a la instancia de producción del VTSS de reserva. Para eliminar esta discrepancia, antes de crear el CDS de prueba de DR, migre la instancia de producción del VTSS de reserva a 0 en el entorno de producción. Puede ejecutar *VTCS VTVRPT OPTION(UNAVAIL)* para garantizar que todos los VTV se migren y estén disponibles para otros VTSS. Si este paso no se realiza, los intentos de la prueba de DR por acceder a los VTV del VTSS de repuesto ejecutarán mensajes SLS6680E a los montajes de VTV.

Recursos de compartidos

Si el entorno de cinta no incluye MVC de VLE o cinta real, puede ejecutar la CDRT en un entorno de VTSS compartido. El parámetro *DRTEST PRIMEPRD/CREATE SHARE* especifica que la producción y la prueba de DR comparten un VTSS durante la prueba. Este entorno aplica las siguientes restricciones:

1. *DRTEST CREATE* no puede definir también recursos DRACS o STORMNGR. Es decir, no se pueden mirar datos del VTSS compartido a medios externos.
2. El sistema de producción debe definir recursos de volúmenes mediante la utilidad *POOLPARM/VOLPARM*.

3. Los montajes de un VTV de una subagrupación distinta de DRTEST en el entorno de prueba de DR hacen que el VTV sea de solo lectura. Es decir, no se permite que la prueba de DR se sobrescriba o se agregue a ningún VTV de producción.

Recursos de ACS

Los recursos de ACS se utilizan para los escenarios de prueba de DR con cinta real únicamente o con cinta virtual sin VLE en un entorno de VSM sin cinta. Los recursos de ACS para CDRT se especifican en el parámetro *DRTEST PRIMEPRD/CREATE DRACS*.

Restricciones de ACS sobre la prueba de DR

Debe ejecutar el comando *CAPPREF* de HSC para establecer los CAP en modo manual antes de ejecutar la utilidad *DRTEST CREATE*. El software garantiza que permanezcan en ese estado durante la realización de la prueba. Después de que introduce el comando *DRTEST START*, se aplican restricciones de ACS sobre la prueba de DR, tanto en el entorno de producción como en el de prueba de DR, para garantizar la coherencia en el hardware y los respectivos CDS a fin de permitir que ambos sitios accedan a los datos. El comando *DRTEST START* se rechaza si un CAP en un ACS de prueba de DR está en modo automático en el entorno de producción.

El software aplica automáticamente las siguientes restricciones.

Requisitos del host de producción de DR

Los requisitos del host de producción de DR durante una prueba de DR activa para el ACS de DR son los siguientes:

- Los CAP deben permanecer en modo manual.
- *FLOAT(OFF)* y *EJCTAUTO(OFF)* se aplican automáticamente, independientemente de las configuraciones del comando MNTD.
- No puede ejecutar las utilidades de expulsión, movimiento, auditoría y reutilización para el ACS de prueba de DR.

Requisitos del host de prueba de DR

Los requisitos del host de prueba de DR durante una prueba de DR son los siguientes:

- Ningún ACS distinto de los de prueba de DR se puede cambiar al estado en línea.
- Los CAP deben permanecer en modo manual.
- *FLOAT(OFF)* y *EJCTAUTO(OFF)* se aplican automáticamente y ningún otro valor es válido en el comando MNTD.
- No puede ejecutar las utilidades de expulsión, movimiento, auditoría y reutilización para el ACS de prueba de DR.

- Solo se permiten actualizaciones reutilizables para volúmenes definidos mediante *POOLPARM/VOLPARM* como volúmenes de subagrupaciones reutilizables de prueba de DR.

Recursos de VLE

Puede definir recursos de VLE mediante el parámetro *STORMNGR* de los comandos *PRIMEPRD* y *CREATE* de DRTEST. Por lo general, los recursos de prueba de DR son los ACS o las VLE, aunque no hay ninguna restricción sobre el uso de ambos.

Al igual que los MVC físicos, los VMVC de VLE se definen mediante la utilidad *POOLPARM/VOLPARM*, con un rango de VMVC reservado para la prueba de DR. La prueba de DR también tiene acceso de lectura a los VMVC de producción, como se muestra en el siguiente ejemplo.

```
//ACTMVCG1 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: MVCMAINT READONLY(ON) STATEMENTS
//SLUSMVON DD DSN=h1q.SLUSMVON, DISP=(NEW, CATLG, DELETE),
// SPACE=(CYL, 1)
/* NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUSMVOF DD DSN=h1q.SLUSMVOF, DISP=(NEW, CATLG, DELETE),
SPACE=(CYL, 1)
/* NOTE: THE FOLLOWING STEP SELECTS ALL "ACTIVE" MVCS
/* IN VLE1 AND MVCPool MVCP1.
//SLSIN DD *
ACTMVCGN STORMNGR=VLE1, MVCP=MVCP1
/*
//ACTMVCG2 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: EXEC MVCMAINT TO SET READONLY(ON)
//SLSIN DD DSN=h1q.SLUSMVON, DISP=SHR
```

Políticas de VTCS

Para minimizar las diferencias entre los entornos de producción y de prueba de DR, la CDRT le permite definir políticas para la gestión de VTV que tienen el mismo nombre en los entornos de producción y prueba de DR, pero distintas definiciones de políticas.

Tenga en cuenta que los sitios de producción y de prueba pueden compartir un VTSS mediante el parámetro *DRTEST DRVTSS SHARE*. También se debe tener en cuenta que ya no es necesario especificar un ACS ficticio de DR para un VTSS compartido o una prueba de DR que utiliza la VLE. La especificación de un VTSS compartido presenta las siguientes restricciones:

- Un VTSS compartido de prueba de DR no debe tener conexiones de RTD activas del sitio de producción o de prueba de DR.
- El CDS debe incluir definiciones *VOLPARM* para definir las subagrupaciones reutilizables de prueba de DR.

- Los VTV de producción (los que no están en una subagrupación de prueba de DR) se deben definir como de solo lectura cuando se montan, y no se pueden modificar.

Definición de MGMTCLAS/STORCLAS para VTSS no compartidos

Cuando define clases de gestión para un sistema de prueba de DR, normalmente, define solamente una copia de MVC de los VTV de salida. Para permitir la limpieza del VTSS después de la prueba, se recomienda que se especifique *DELSCR(YES)*, como se muestra en el siguiente ejemplo.

```
STOR NAME(LOCAL) ACS(01) MVCP00L(MVCP1)
MGMT NAME(CRITICAL) MIGPOL(LOCAL) IMMWAIT(0) DELSCR(YES)
```

Definición de MGMTCLAS para VTSS compartidos

Cuando usa un VTSS compartido para la prueba de DR, no se permiten copias de salida migradas. Debe especificar *DELSCR(YES)* para permitir la limpieza después de la prueba, como se muestra en el siguiente ejemplo.

```
MGMT NAME(CRITICAL) DELSCR(YES)
```

Optimización del acceso a recursos de prueba y producción

Durante una prueba de DR, se recomienda que aplique procedimientos para optimizar el acceso a los recursos, tanto en el entorno de prueba como en el de producción. Específicamente:

- Antes de comenzar la prueba de DR, defina clases de gestión de producción que especifiquen la migración inmediata a los ACS de producción y prueba de DR, de modo que los VTV estén disponibles en los MVC accesibles para el sistema de prueba de DR y el sistema de producción.
- Defina clases de gestión de prueba de DR que especifiquen una sola copia de migración, dado que un solo ACS está normalmente disponible para el sitio de prueba de DR.
- Use la utilidad *POOLPARAM/VOLPARAM* para separar ambas subagrupaciones reutilizables y las agrupaciones de MVC entre la producción y la prueba de DR.
- De ser posible, asegúrese de que el procesamiento de la prueba de DR no actualice ningún VTV preexistente (*DISP=MOD* o sobrescritura con *DISP=OLD*).
- Minimice la contención entre los trabajos de producción que migran al ACS de prueba de DR y los trabajos de prueba de DR que acceden a los VTV de los MVC en el ACS de prueba de DR mediante la ejecución de la utilidad *ACTMVCGN* para marcar los MVC activos como de solo lectura en el entorno de producción.
- Desactive la recuperación de espacio del MVC (mediante *CONFIG HOST NORECLAM*) en los MVC de producción durante la prueba de DR, a fin de preservar el contenido de volúmenes de MVC que está utilizando el sistema de prueba de DR.

Ejecución de una prueba de DR

Nota:

Para obtener más información acerca de los comandos y las utilidades que se usan en este procedimiento, consulte la *Referencia de comandos, sentencia de control y utilidades de ELS*.

Para ejecutar una prueba de DR:

1. Defina agrupaciones de volúmenes en el CDS de producción mediante el comando *SET VOLPARM* y las sentencias *POOLPARM/VOLPARM* en *SLSPARM DD*.

Para obtener más información, consulte " [Recursos de volúmenes](#)."

2. Asegúrese de que los recursos de prueba estén correctamente definidos.

Para obtener más información, consulte:

- "Recursos de ACS"
- "Recursos de VTSS"
- "Recursos de VLE"

3. Cree sentencias *MGMTCLAS/STORCLAS* para el entorno DRTEST.

Para obtener más información, consulte:

- "Definición de *MGMTCLAS/STORCLAS* para VTSS no compartidos"
- "Definición de *MGMTCLAS* para VTSS compartidos"

4. Si es necesario, copie el catálogo de MVS para el sitio de prueba de DR.
5. De manera opcional, copie la base de datos de TMS (si se usa un TMS) para el sitio de prueba de DR.
6. En el sitio de producción, ejecute la utilidad DRTEST (mediante la palabra clave *PRIMEprd*) para preparar el CDS de producción para la prueba de DR.

Por ejemplo, consulte " [JCL de muestra para el escenario 1](#)."

Debe ejecutar *PRIMEprd* **una vez** en el entorno, independientemente de cuántas iteraciones de DRTEST ejecute, **a menos** que la configuración cambie. Si la configuración de la prueba de DR cambia de alguna manera, debe volver a ejecutar *PRIMEprd*. Tenga en cuenta también que no debe ejecutar la utilidad *DRTEST RESET* después de que la prueba de DR se ha completado. Si bien los indicadores permanecen establecidos en el CDS de producción, no afectan el procesamiento, siempre que la prueba de DR no esté activa.

7. En el sistema de producción, use el comando *CAPPREF* de HSC para establecer todos los CAP del ACS de prueba de DR en modo manual.
8. En el sitio de prueba de DR, ejecute la utilidad DRTEST (mediante la palabra clave *CREATE*) en una copia de seguridad o reflejada del CDS de producción para crear el nuevo CDS de prueba de DR para la prueba de DR.

Cada escenario ofrece un ejemplo de *DRTEST CREATE*.

Los CDS se deben asignar utilizando las sentencias DD en la utilidad. Tenga en cuenta que cuando se utiliza *NOUPD*, solo se necesita la sentencia *SLSCNTL DD*, y puede tratarse del CDS principal real, una copia de seguridad o una copia reflejada.

9. Comience la prueba de DR en el sitio de producción, apunte a las definiciones *MGMTCLAS/STORCLAS* de *DRTEST* que creó en el paso 3.

Por ejemplo:

```
/PRIME EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSIN DD *
DRTEST START
```

Nota:

También puede iniciar la prueba si introduce el comando *DRTEST START* desde la consola.

10. Inicie el sistema SMC en los hosts del cliente de *DRTEST*.
11. Inicie los sistemas SMC/HSC/VTCS (apunte a los CDS que creó en el paso 8) en los sistemas de prueba de DR.
12. Verifique que los VTD para los VTSS de prueba y las rutas estén en línea.
13. Cambie los VTSS de DR a en línea para el sistema de DR.
14. Si corresponde, cambie las RTD de DR a en línea para el sistema de DR.
15. Ejecute las pruebas en el sitio de prueba de DR.

Durante la prueba de DR, se aplican las siguientes condiciones programáticas:

- Los ACS del sitio de producción se desconectan de los hosts de prueba de DR.
- Los VTSS del sitio de producción están fuera de línea para los hosts de prueba de DR.
- No se pueden producir desmontajes, expulsiones, movimientos, actualizaciones reutilizables, auditorías ni redistribuciones reutilizables flotantes en el sitio de prueba de DR.
- No se pueden producir desmontajes, inserciones/expulsiones, movimientos, auditorías ni redistribuciones reutilizables flotantes en el ACS de prueba de DR en el sitio de producción.
- Todos los CAP del ACS de prueba de DR se encuentran en modo manual.

Nota:

Puede introducir volúmenes en el ACS de prueba de DR, pero una vez que la prueba ha finalizado, debe expulsar los volúmenes o auditar las celdas para sincronizar el CDS de producción con los volúmenes de biblioteca reales.

Limpeza después de una prueba de DR

Como se describe al comienzo de este capítulo, es fundamental que los trabajos de un flujo de trabajo de prueba de DR no actualicen los volúmenes creados en la producción.

Nota:

Para obtener información acerca del comando *DRTEST* y la utilidad *DRTEST*, consulte la *Referencia de comandos, sentencia de control y utilidades de ELS*. Para obtener información acerca de los mensajes de CDRT, consulte los *Mensajes y códigos de ELS*.

Limpeza después de una prueba de DR

Nota:

Para ELS 7.1 y superiores, si tiene instalado SPE de mejora de limpieza de CDRT, no debe ejecutar este procedimiento inmediatamente después de la prueba de DR o antes de reanudar el entorno de producción normal. En cambio, puede ejecutarlo sin interrupciones (por ejemplo, antes de la próxima prueba de DR).

1. Ejecute la utilidad *SCRATCH* de SLUADMIN para reutilizar todos los VTV posibles en las agrupaciones de DRTEST.

Debido a que establece *DELSCR(YES)* en la clase de gestión, los VTV se suprimirán automáticamente del buffer cuando los reutilice al final de la prueba.

ADVERTENCIA:

Si no utiliza SET VOLPARM y no establece agrupaciones reutilizables independientes, correrá el riesgo de perder datos.

2. Si la prueba de DR ha modificado o puede haber modificado cualquier VTV de producción, también debe hacer lo siguiente para asegurarse de que ningún dato de prueba de DR permanezca en el VTSS de producción:
 - Ejecute un informe de VTV en el CDS de DRTEST y examine la salida para determinar si algún VTV en los rangos de VTV de producción se modificaron durante la prueba.

Tenga en cuenta que *VTVRPT COPIES* ahora indica con una "D" en la columna de DRT las copias de VTV que son copias de prueba de DR.
 - Si se modificó un VTV, debe realizar alguna de las siguientes acciones:
 - Realice una migración bajo demanda de los VTV modificados en función del informe de VTV.
 - Migre el VTSS de prueba de DR a 0.
 - Solicítele a un CSE que "limpie" el VTSS de prueba.
3. Detenga el HSC/VTCS y el SMC en el sistema de MVC de PRUEBA de DR.
4. Si la utilidad *ACTMVCGN* se ejecutó antes de la prueba para colocar los MVC en el estado *READONLY*, ejecute SLUADMIN mediante la salida de la sentencia *SLUSMVOF DD* como entrada para restablecer el estado *READONLY*.

Reanudación de las operaciones normales

Siga este procedimiento para reiniciar las operaciones y detener la prueba de DR.

1. Detenga la prueba de DR en el sistema MVS de *PRODUCTION*.

Por ejemplo:

```
/STOP EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRTEST STOP
```

Tenga en cuenta también que no debe ejecutar la utilidad *DRTEST RESET* después de que la prueba de DR se ha completado. Si bien los indicadores permanecen establecidos en el CDS de producción, no afectan el procesamiento, siempre que la prueba de DR no esté activa.

2. Si lo desea, coloque los CAP en modo automático.

Escenarios operativos

En esta sección, se indica cómo usar el software de prueba de DR para configurar el entorno para iniciar y detener las pruebas de DR. Esta sección incluye la siguiente información:

- "Escenario 1: Sitios de prueba y producción, ACS en cada sitio, VTSS de reserva en el sitio de prueba"
- "Escenario 2: Sitios de prueba y producción, ACS en cada sitio, toma de control del VTSS en el sitio de prueba"
- "Escenario 3: Sitios de prueba y producción, ACS en cada sitio, sin VTSS"
- "Escenario 4: VTSS en cluster con sitios de producción y prueba de DR"
- "Escenario 5: Sitios de prueba y producción, ACS y VLE en cada sitio"
- "Escenario 6: Sitios de prueba y producción, solo VLE en cada sitio"
- "Escenario 7: VTSS en cluster (sin cinta) con sitios de producción y prueba de DR"

Para obtener información acerca del comando *DRTEST* y la utilidad *DRTEST*, consulte la *Referencia de comandos, sentencias de control y utilidades de ELS*. Para obtener información acerca de los mensajes de *CDRT*, consulte los *Mensajes y códigos de ELS*.

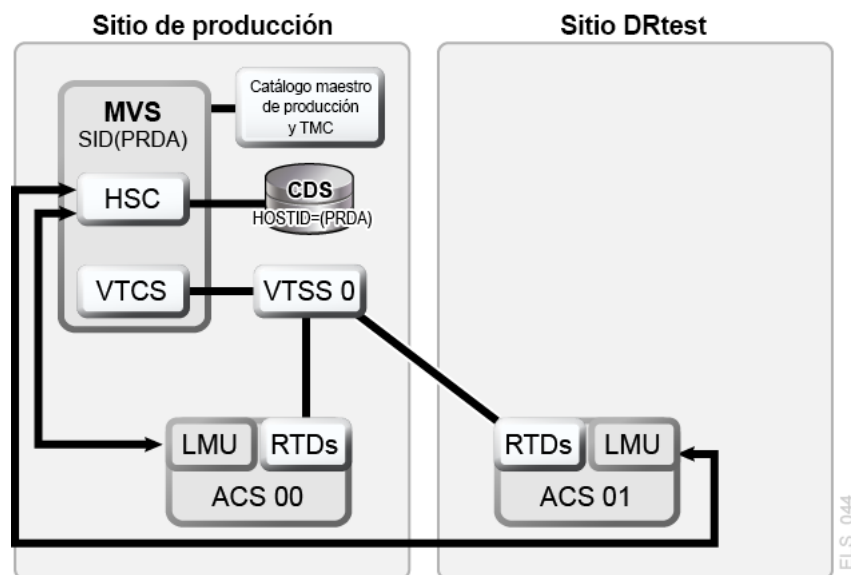
Nota:

Para todos los escenarios, después de la prueba, ejecute el procedimiento que se indica "[Limpieza después de una prueba de DR.](#)"

Escenario 1: Sitios de prueba y producción, ACS en cada sitio, VTSS de reserva en el sitio de prueba

En el escenario 1, hay un solo ACS en los sitios de prueba y producción, y los VTSS de "reserva" en el sitio de prueba se usan únicamente para la realización de pruebas (no hay requisitos que establezcan que se debe migrar o restaurar el contenido de los VTSS de "reserva"). En las operaciones normales, el sitio de producción escribe los VTV y accede a ellos en los VTSS del sitio de producción, y los VTV de salida siempre se migran inmediatamente y se duplican en MVC separados, uno en cada ACS. En [Figura 7.1, "Configuración de VTSS de reserva: antes de la ejecución de la utilidad DRTEST"](#), se muestra el sistema del escenario 1 antes de ejecutar la utilidad DRTEST.

Figura 7.1. Configuración de VTSS de reserva: antes de la ejecución de la utilidad DRTEST



JCL de muestra para el escenario 1

Paso PRIMEPRD:

```
//DRTPRIME EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRTEST PRIMEPRD +
DRACS(01) DRVTSS(VTSS0) SPARE HOST(MVS1,MVS2)
```

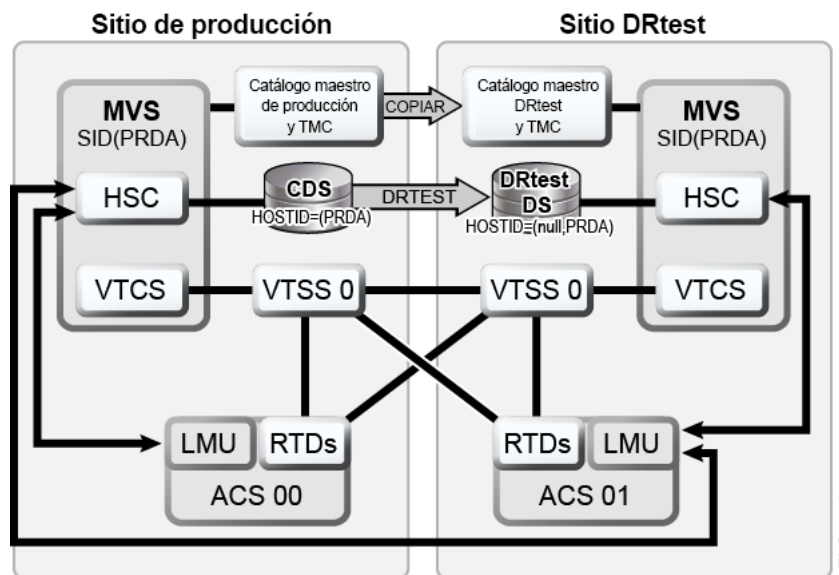
Paso CREATE:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
```

```
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRACS(01) DRVTSS(VTSS0) SPARE HOST(MVS1, MVS2)
```

En Figura 7.2, “Configuración de VTSS de reserva: después de la ejecución de la utilidad DRTEST”, se muestra el sistema del escenario 1 (VTSS de reserva en el sitio de prueba) después de la ejecución de la utilidad DRTEST.

Figura 7.2. Configuración de VTSS de reserva: después de la ejecución de la utilidad DRTEST



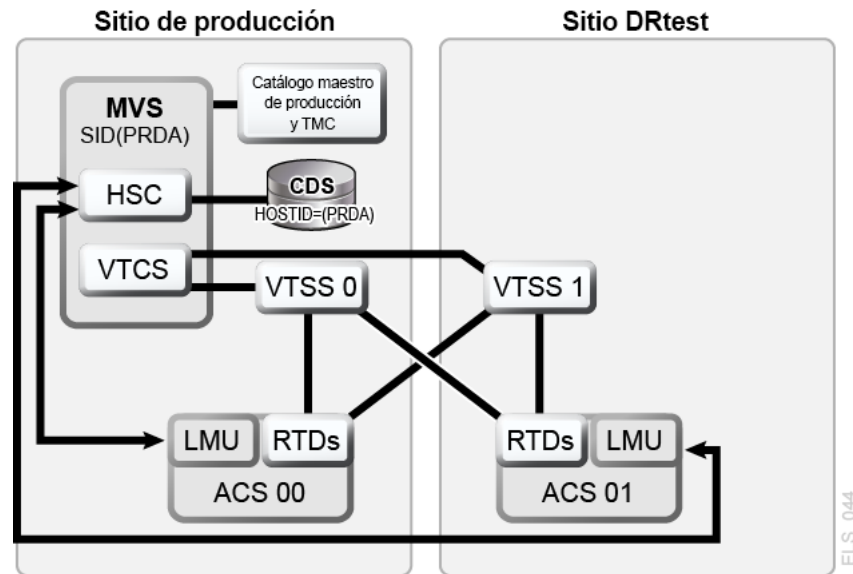
Escenario 2: Sitios de prueba y producción, ACS en cada sitio, toma de control del VTSS en el sitio de prueba

En el escenario 2, hay un solo ACS, tanto en el sitio de producción como en el de prueba, pero no hay VTSS de "reserva" en el sitio de prueba para la realización de las pruebas. En las operaciones normales, el sitio de producción escribe los VTV y accede a ellos en los VTSS de ambos sitios, y los VTV de salida siempre se migran inmediatamente y se duplican en MVC separados, uno en cada ACS. En esta configuración, debe realizar una migración bajo demanda a cero de uno o más VTSS en el sitio de prueba y colocar estos VTSS en estado fuera de línea para el sistema de producción, de modo que las pruebas puedan tomar el control de los recursos de VTSS necesarios. Además, en el sitio de prueba, una o más LPAR

funcionan como sistemas de producción desplazados que se ejecutan en paralelo con los sistemas de producción reales. Ambos ACS están en línea para el sitio de producción.

En [Figura 7.3, “Configuración de toma de control del VTSS: antes de la ejecución de la utilidad DRTEST”](#), se muestra el sistema del escenario 2 (toma de control del VTSS en el sitio de prueba) antes de la ejecución de la utilidad *DRTEST*.

Figura 7.3. Configuración de toma de control del VTSS: antes de la ejecución de la utilidad DRTEST



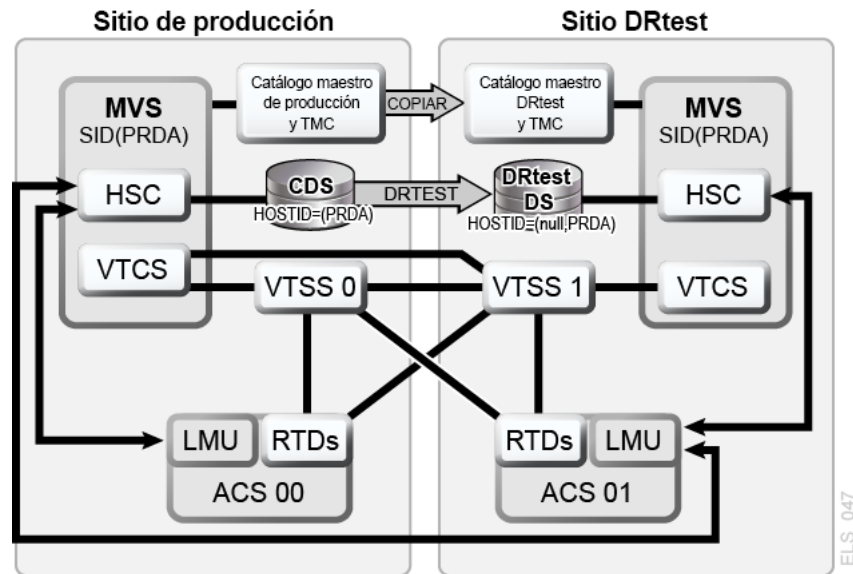
JCL de muestra para el escenario 2

Paso CREATE solamente, PRIMEPRD ejecutado anteriormente:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRACS(01) DRVTSS(VTSS1) HOST(MVS1, MVS2)
```

En [Figura 7.4, “Configuración de toma de control del VTSS: después de la ejecución de la utilidad DRTEST”](#), se muestra el sistema del escenario 2 (toma de control del VTSS en el sitio de prueba) después de la ejecución de la utilidad *DRTEST*.

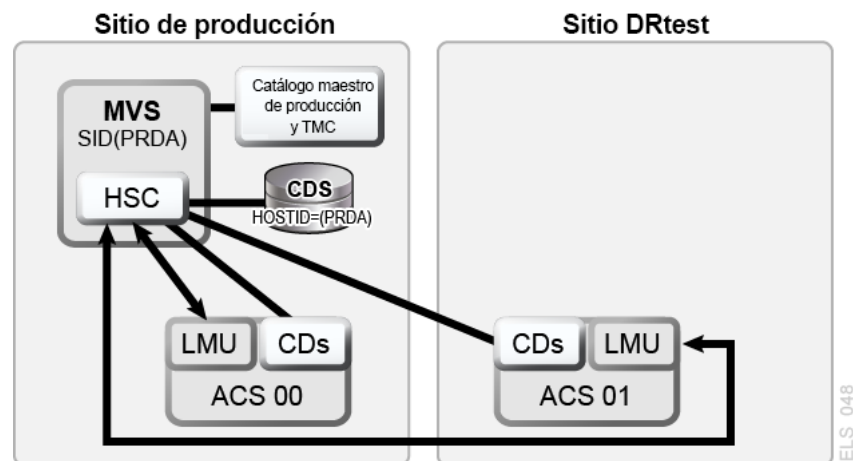
Figura 7.4. Configuración de toma de control del VTSS: después de la ejecución de la utilidad DRTEST



Escenario 3: Sitios de prueba y producción, ACS en cada sitio, sin VTSS

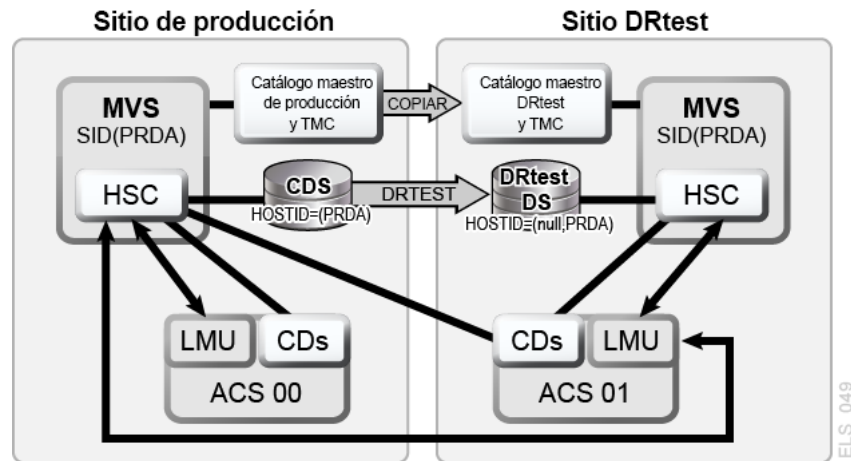
En el escenario 3, hay un solo ACS, tanto en el sitio de producción como en el de prueba, pero no hay ningún VTSS en el sitio de prueba para la realización de las pruebas. En las operaciones normales, el sitio de producción escribe los juegos de datos de cinta y accede a ellos en ambos sitios. En [Figura 7.5, “Configuración real únicamente: antes de la ejecución de la utilidad DRTEST”](#), se muestra el sistema del escenario 3 (configuración real únicamente) antes de la ejecución de la utilidad *DRTEST*.

Figura 7.5. Configuración real únicamente: antes de la ejecución de la utilidad DRTEST



En [Figura 7.6, “Configuración real únicamente: después de la ejecución de la utilidad DRTEST”](#), se muestra el sistema del escenario 3 (configuración real únicamente) después de la ejecución de la utilidad *DRTEST*.

Figura 7.6. Configuración real únicamente: después de la ejecución de la utilidad DRTEST



JCL de muestra para el escenario 3

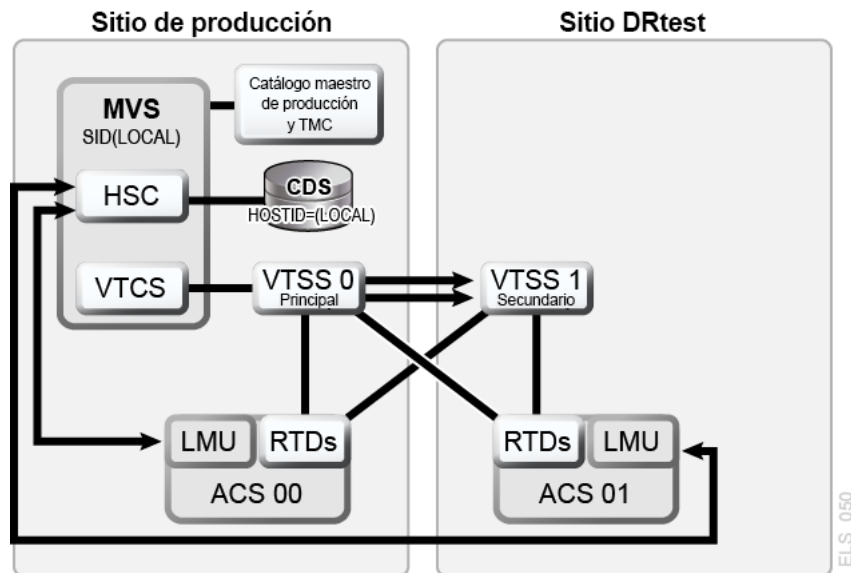
Paso CREATE solamente, PRIMEPRD ejecutado anteriormente:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDRD +
DRACS(01) HOST(MVS1, MVS2)
```

Escenario 4: VTSS en cluster con sitios de producción y prueba de DR

Como se muestra en [Figura 7.7, “Configuración de un VTSS principal/secundario en cluster: operaciones normales”](#), en las operaciones normales, el escenario 4 es una configuración de VTSS en cluster que se usa para DR, con los sitios de producción y prueba de DR interconectados a los ACS de producción y prueba de DR. En el sitio de producción, VTSS0 es el principal y VTSS1 es el secundario en el sitio de prueba de DR.

Figura 7.7. Configuración de un VTSS principal/secundario en cluster: operaciones normales



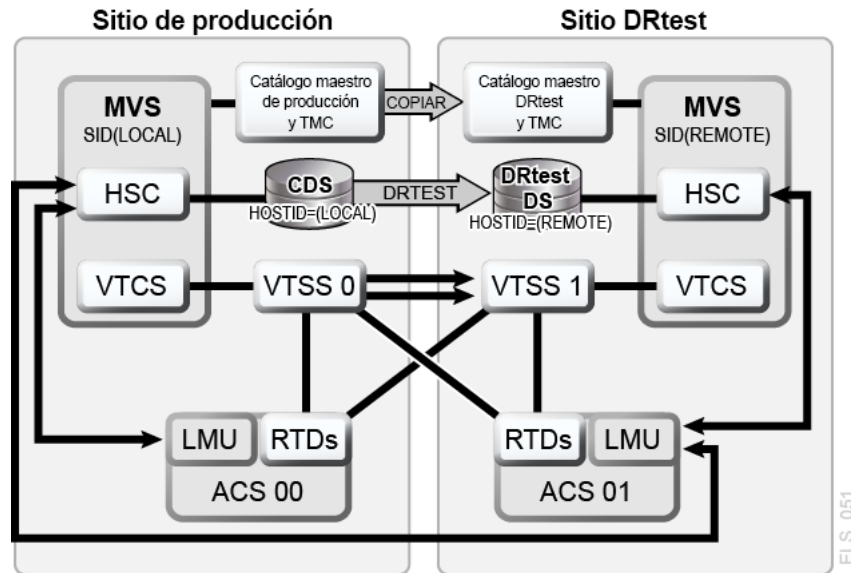
JCL de muestra para el escenario 4

Paso CREATE solamente, PRIMEPRD ejecutado anteriormente:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM=' MIXED '
//STEPLIB DD DSN=h1q. SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q. DRTEST. SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q. DRTEST. SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q. DRTEST. SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRACS(01) DRVTSS(VTSS1) SHARE HOST(MVS1, MVS2)
```

¿Qué ocurre si desea usar el sitio de prueba de DR para realizar pruebas? En [Figura 7.8, “Configuración de un VTSS principal/secundario en cluster: durante la prueba”](#), se muestra el sistema del escenario 4 durante la prueba de DR.

Figura 7.8. Configuración de un VTSS principal/secundario en cluster: durante la prueba

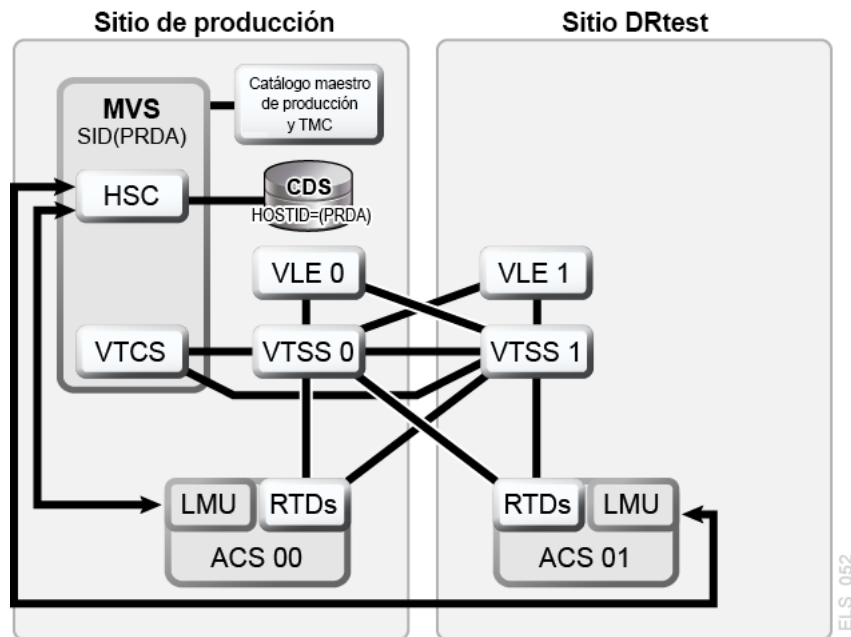


Escenario 5: Sitios de prueba y producción, ACS y VLE en cada sitio

En el escenario 5, hay un solo ACS, tanto en el sitio de producción como en el de prueba, pero no hay VTSS de "reserva" en el sitio de prueba para la realización de las pruebas. En las operaciones normales, el sitio de producción escribe los VTV y accede a ellos en los VTSS de ambos sitios, y los VTV de salida siempre se migran inmediatamente y se duplican; una copia se migra a un MVC del ACS y otra, a un VMVC de la VLE. En esta configuración, debe realizar una migración bajo demanda a cero de uno o más VTSS en el sitio de prueba y colocar estos VTSS en estado fuera de línea para el sistema de producción, de modo que las pruebas puedan tomar el control de los recursos de VTSS necesarios. Además, en el sitio de prueba, una o más LPAR funcionan como sistemas de producción desplazados que se ejecutan en paralelo con los sistemas de producción reales. **Tanto los ACS como las VLE están en línea para el sistema de producción.**

En [Figura 7.9, "Configuración de VLE y ACS: antes de la ejecución de la utilidad DRTEST"](#), se muestra el sistema del escenario 5 antes de la ejecución de la utilidad *DRTEST*.

Figura 7.9. Configuración de VLE y ACS: antes de la ejecución de la utilidad DRTEST



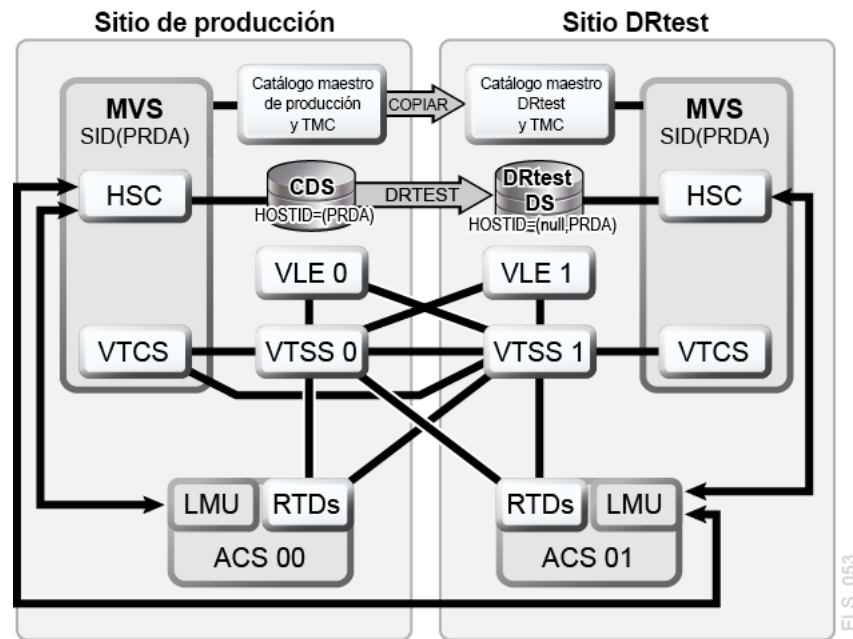
JCL de muestra para el escenario 5

Paso CREATE solamente, PRIMEPRD ejecutado anteriormente:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM=' MIXED '
//STEPLIB DD DSN=h1q. SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q. DRTEST. SLSCNTL, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q. DRTEST. SLSCNTL2, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q. DRTEST. SLSSTBY, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDRD +
DRACS(01) DRVTSS(VTSS1) HOST(MVS1, MVS2) STORMNGR(VLE1)
```

En [Figura 7.10, “Configuración de VLE y ACS: después de la ejecución de la utilidad DRTEST”](#), se muestra el sistema del escenario 5 después de la ejecución de la utilidad *DRTEST*.

Figura 7.10. Configuración de VLE y ACS: después de la ejecución de la utilidad DRTEST



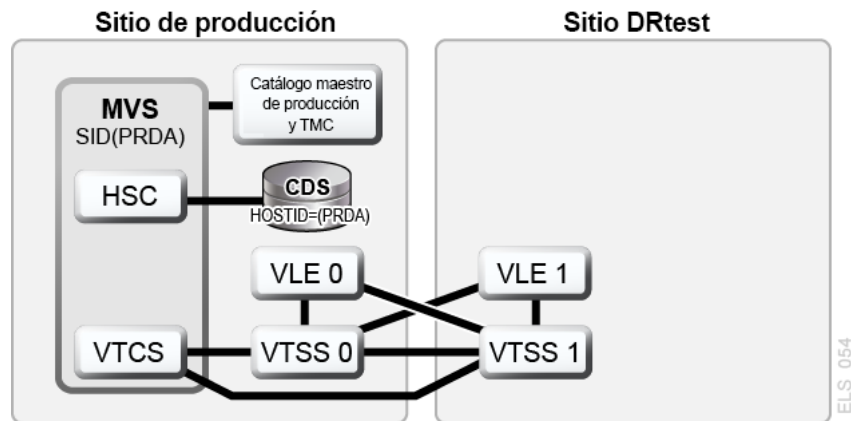
Escenario 6: Sitios de prueba y producción, solo VLE en cada sitio

En el escenario 6, hay un solo VTSS con una VLE conectada en cada sitio. El VTSS del sitio de prueba **no** es de reserva y lo usa el sitio de producción durante las operaciones normales. Los VTSS de salida siempre se migran inmediatamente y se duplican en VMVC independientes, uno en cada VLE.

En esta configuración, debe realizar una migración bajo demanda a cero de uno o más VTSS en el sitio de prueba y colocar estos VTSS en estado fuera de línea para el sistema de producción, de modo que las pruebas puedan tomar el control de los recursos de VTSS necesarios. Además, en el sitio de prueba, una o más LPAR funcionan como sistemas de producción desplazados que se ejecutan en paralelo con los sistemas de producción reales. Ambas VLE están en línea para el sitio de producción.

En [Figura 7.11, "Configuración de solo VLE: antes de la ejecución de la utilidad DRTEST"](#), se muestra el sistema del escenario 6 antes de la ejecución de la utilidad *DRTEST*.

Figura 7.11. Configuración de solo VLE: antes de la ejecución de la utilidad DRTEST



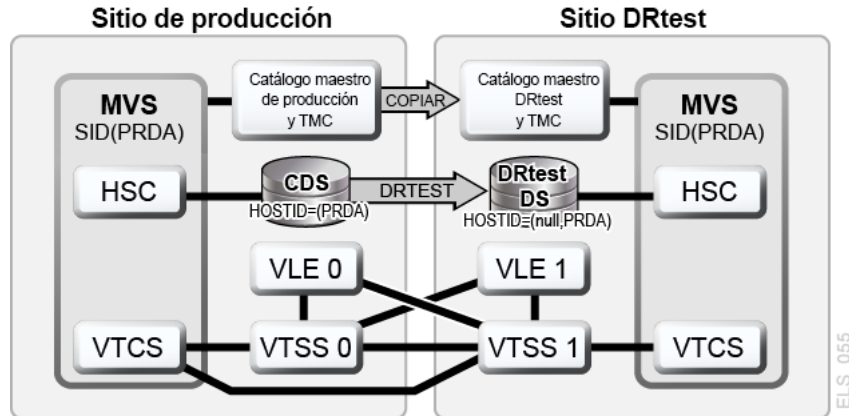
JCL de muestra para el escenario 6

Paso CREATE solamente, PRIMEPRD ejecutado anteriormente:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDRD +
STORMNGR(VLE1) DRVTSS(VTSS1) HOST(MVS1, MVS2)
```

En [Figura 7.12](#), “Escenario de solo VLE: después de la ejecución de la utilidad DRTEST”, se muestra el sistema del escenario 6 después de la ejecución de la utilidad *DRTEST*.

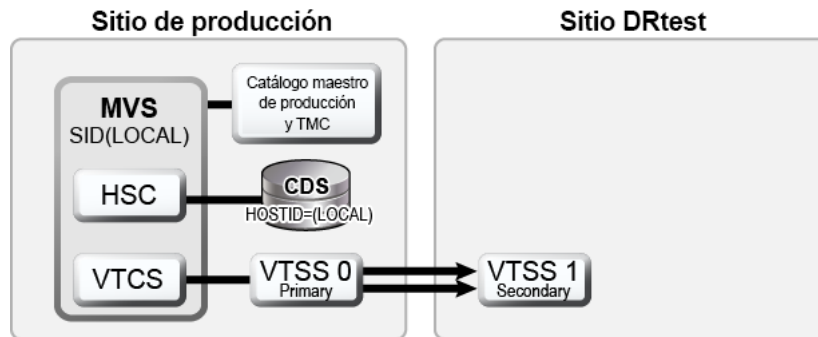
Figura 7.12. Escenario de solo VLE: después de la ejecución de la utilidad DRTEST



Escenario 7: VTSS en cluster (sin cinta) con sitios de producción y prueba de DR

Como se muestra en [Figura 7.13, “Configuración de un VTSS principal/secundario en cluster sin cinta: antes de la ejecución de la utilidad DRTEST”](#), en las operaciones normales, el escenario 7 es una configuración de VTSS en cluster (sin cinta) utilizada para DR, con un sitio de producción y uno de prueba de DR. En el sitio de producción, VTSS0 es el principal y VTSS1 es el secundario en el sitio de prueba de DR.

Figura 7.13. Configuración de un VTSS principal/secundario en cluster sin cinta: antes de la ejecución de la utilidad DRTEST



JCL de muestra para el escenario 7

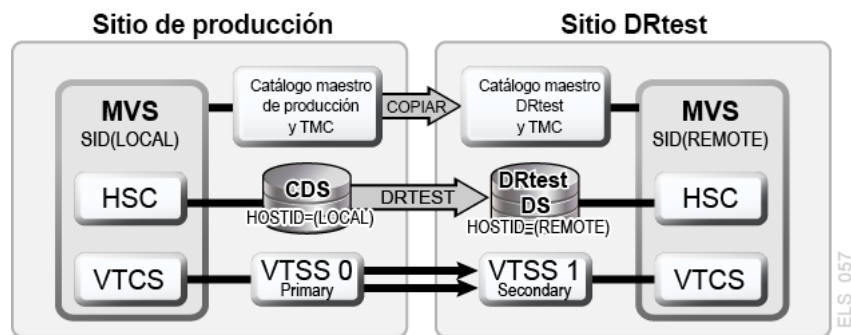
Paso CREATE solamente, PRIMEPRD ejecutado anteriormente:

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
```

```
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
//          UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRVTSS(VTSS1) SHARE HOST(MVS1, MVS2))
```

¿Qué ocurre si desea usar el sitio de prueba de DR para realizar pruebas? En [Figura 7.14](#), “Configuración de un VTSS principal/secundario en cluster sin cinta: durante la prueba”, se muestra el sistema del escenario 7 durante la prueba de DR.

Figura 7.14. Configuración de un VTSS principal/secundario en cluster sin cinta: durante la prueba



Capítulo 8. Creación de puntos de recuperación del sistema en entornos de VSM

Como se describe en " [Definición del objetivo de punto de recuperación \(RTO\)](#)," uno de los aspectos más importantes de una solución de DR exitosa es la capacidad de establecer puntos de control del sistema que garanticen que un juego coherente de datos se pueda usar como punto de partida de DR.

En relación con los entornos de VSM, un punto de partida válido de DR es aquel donde:

- Todos los datos empresariales críticos están protegidos en la ubicación de DR designada.
- Se ha capturado una copia segura de los metadatos (CDS, catálogo de MVS, TMC).
- Se garantiza una copia válida de los metadatos cuando se declara un desastre (ya sea un desastre real o una prueba).

El VTCS ofrece la capacidad de crear un punto de partida de DR mediante las siguientes funciones:

- La utilidad *DRMONitr* supervisa los datos críticos de DR y garantiza que lleguen a la ubicación de recuperación designada. Permite que el procesamiento del flujo de trabajo se detenga, a la espera de que los datos lleguen a su destino. Una vez que se contabilizan todos los datos, la utilidad termina. La utilidad *DRMONitr* se puede ejecutar como paso de trabajo. Al finalizar el paso de trabajo, se garantiza que todos los datos supervisados se han contabilizado y se encuentran protegidos en la ubicación de DR designada.
- La utilidad *DRCHKPT* se utiliza para garantizar que los datos a los que se acceda mediante los metadatos del CDS sigan siendo válidos durante un período establecido. Esto garantiza que una copia de seguridad del CDS siga siendo válida durante un período establecido y, por lo tanto, puede restaurar un sistema VSM nuevamente a un punto de partida de DR. La utilidad *DRCHKPT* establece un registro de fecha y hora en el CDS activo, que establece el punto de recuperación a partir del cual el contenido de MVC se puede recuperar. A partir de este punto temporal de recuperación, se garantiza el contenido de los datos durante algún tiempo. Sin la utilidad *DRCHKPT*, una copia de seguridad del CDS no se puede usar para restaurarla nuevamente al punto de partida de DR, ya que es posible que los elementos del CDS (posición de VTV en un MVC) ya no sean válidos.

Para obtener más información, consulte la *Referencia de comandos, sentencia de control y utilidades de ELS*.

También, tenga en cuenta lo siguiente:

- Para los VMVC, *MVCDRAIN* con el parámetro *EJECT* elimina físicamente los VTV.

Atención:

Si usa la utilidad *DRCHKPT* o el parámetro *CONFIG GLOBAL PROTECT* para proteger el contenido de copia de seguridad del CDS para los VMVC, la especificación de *MVCDR EJECT* invalida el contenido del VMVC de la copia de seguridad del CDS.

- Para ambos VMVC y MVC, *MVCDRAIN* sin el parámetro *EJECT* no suprime los VTV, pero actualiza el registro del CDS para no mostrar VTV en el VMVC/MVC.

Para obtener más información, consulte la *Referencia de comandos, sentencia de control y utilidades de ELS*.

Ejemplos de puntos de control

Se explican los siguientes ejemplos:

- ["Ejemplo 1: copias de MVC locales y copias de MVC remotas"](#)
- ["Ejemplo 2: uso de CONFIG RECLAIM PROTECT"](#)

Ejemplo 1: copias de MVC locales y copias de MVC remotas

En este ejemplo:

- Las utilidades *DRMONitr* y *DRCHKPT* garantizan que los datos de DR lleguen a su ubicación de recuperación y que los metadatos asociados (copia de seguridad del CDS) recuperen los datos del VTV de ser necesario.
- Se muestra un sitio local con un VTSS más un ACS (ACS 00) y un sitio remoto con un solo ACS (ACS 01), como se puede ver en [Figura 8.1, "Ejemplo de puntos de recuperación del sistema VSM \(locales y remotos\)"](#).

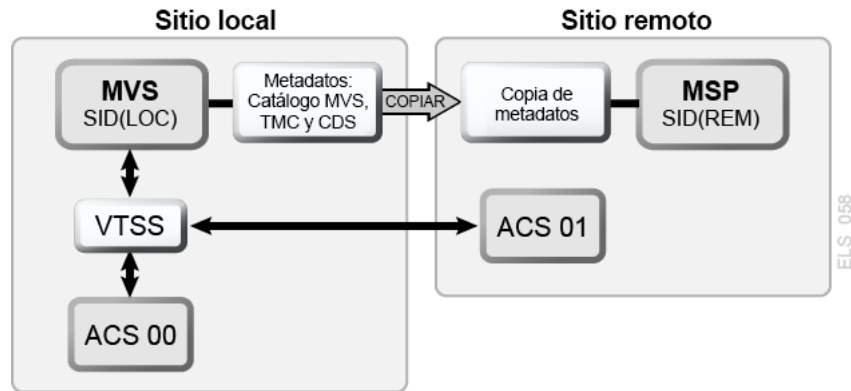
El ejemplo es una estrategia simple de DR donde, diariamente, se protegen copias de datos críticos en el sitio remoto junto con los metadatos. Las copias del VTV remoto son las copias de DR designadas.

Después de que finalizan los trabajos de producción, se programa un trabajo para realizar lo siguiente:

- Supervisar que se hayan completado las copias remotas (*DRMONitr*).
- Verificar los puntos de control del CDS (*DRCHKPT*).
- Realizar una copia de seguridad de los metadatos (CDS, TMC, catálogo de MVS) y protegerla en el sitio remoto. **Se debe tener en cuenta que** las copias de seguridad de los metadatos son clave para la DR; se supone que se trasladan a una ubicación "conocida" o que su ubicación se encuentra registrada en una ubicación segura.

Esto proporciona un punto de control de DR sincronizado de forma diaria. En el caso de que se declare una DR, el entorno de cintas se restaura nuevamente al punto de control, y los trabajos se vuelven a ejecutar a partir de este estado conocido.

Figura 8.1. Ejemplo de puntos de recuperación del sistema VSM (locales y remotos)



Para ejecutar este ejemplo con la configuración que se muestra en [Figura 8.1, “Ejemplo de puntos de recuperación del sistema VSM \(locales y remotos\)”](#):

1. Cree las siguientes sentencias de política.

```
MGMT NAME(DR)  MIGPOL(LOCAL,REMOTE) IMMDELAY(0)
STOR NAME(LOCAL) ACS(00)
STOR NAME(REMOTE) ACS(01)
```

Nota:

Para lograr un entorno de DR eficaz, también se recomienda considerar la posibilidad de usar las sentencias *MIGRSEL* y *MIGRVTV*, que puede usar para garantizar que las copias de DR se protejan tan pronto como sea posible.

2. Para garantizar que los datos críticos se protejan en la ubicación remota, se debe ejecutar el siguiente ejemplo de paso de trabajo de *DRMONitr*.

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SYSIN DD UNIT=SYSDA, SPACE=(TRK, 1)
//*
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRMON MGMT(DR) STOR(REMOTE) MAXAGE(24) TIMEOUT(30)
```

En este ejemplo, la utilidad *DRMONitr* esperará hasta que todas las copias del VTV de DR de clase de gestión que tengan menos de 24 horas se envíen al ACS remoto. La

utilidad se configura para anular la operación si el tiempo de ejecución (o de espera) supera los 30 minutos.

- Una vez que todas las copias del VTV se han entregado al ACS remoto, señalado por un valor de RC igual a cero, *DRCHKPT* se ejecuta para establecer el punto de recuperación, como se muestra en el siguiente ejemplo.

```
//CHKPT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRCHKPT SET
```

En este ejemplo, la utilidad *DRCHKPT* establece un registro de hora o un punto de recuperación en el CDS activo. A partir de este punto temporal de recuperación, se garantiza el contenido de la copia de MVC durante algún tiempo (por ejemplo, hasta que se ejecute otra utilidad *CHKPT*).

- Una vez que el punto de recuperación se establece en el CDS activo, se debe realizar una copia de seguridad del CDS de inmediato, como se muestra en el siguiente ejemplo.

```
//CHKPT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SYSIN DD UNIT=SYSDA, SPACE=(TRK,1)
//*
//SLSCNTL DD DSN=h1q.DBSEPRM, DISP=SHR
//SLSBKUP DD DSN=h1q.DBSEPRM.BKUP, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
BACKUP OPTION(COPY)
```

Después de realizar la copia de seguridad, se garantiza la validez de los metadatos o el contenido del MVC durante algún tiempo (hasta que se establezca un punto de control o de recuperación posterior).

De esta manera, se completa el procedimiento. En caso de que se declare una situación de DR (el sitio de producción local ya no esté disponible) realice una de las siguientes acciones:

- Transporte los MVC y todos los demás datos críticos (copias de metadatos, por ejemplo) a otra ubicación donde esté disponible un reflejo del sitio de producción local.
- Genere una réplica del sitio de producción local en la ubicación remota.

Los metadatos se restauran (CDS, TMC, catálogo de MVS). Al reiniciar el entorno de cintas, todo puede continuar (avanzar) a partir del punto de sincronización de DR.

Ejemplo 2: uso de CONFIG RECLAIM PROTECT

En este ejemplo, se realizan copias de seguridad del CDS cada 24 horas. El contenido del MVC o los metadatos del CDS que se encuentran dentro de la copia de seguridad del CDS deben conservar su validez hasta que se realice una copia de seguridad posterior del CDS.

En este ejemplo, se muestra la protección de MVC establecida en 28 horas. Para obtener más información sobre el parámetro *CONFIG RECLAIM PROTECT*, consulte *Referencia de comandos, sentencias de control y utilidades de ELS*.

1. Defina *CONFIG GLOBAL PROTECT = 28*.
2. El día 1, realice una copia de seguridad del CDS.
 - Los MVC drenados/recuperados después de esta copia de seguridad no se pueden sobrescribir durante 28 horas.
 - Ahora, la copia de seguridad del CDS del día 1 es el punto de recuperación hasta la próxima copia de seguridad del CDS.
3. El día 2, realice una copia de seguridad del CDS.
 - Los MVC drenados/recuperados después de esta copia de seguridad no se pueden sobrescribir durante 28 horas.
 - Ahora, la copia de seguridad del CDS del día 2 es el punto de recuperación hasta la próxima copia de seguridad del CDS.

Capítulo 9. Uso de la VLE para recuperación ante desastres

El uso de la VLE (Virtual Library Extension) como solución de recuperación ante desastres proporciona un método simplificado y sin interrupciones de realización de pruebas de DR y de recuperación a partir de un evento de interrupción de las actividades.

El sistema gestiona la VLE como una biblioteca (ACS). No obstante, debido a que la VLE usa almacenamiento en disco en lugar de almacenamiento en cinta, y a que mantiene un inventario interno de los VTV en su contenido, ofrece funciones que una biblioteca real no puede ofrecer:

- La VLE es una solución "sin cinta", que permite evitar los problemas de la gestión de medios.
- Los datos se envían a la VLE mediante IP y no requieren extensión de canal.
- La VLE puede realizar una auditoría de MVC en cuestión de segundos mediante su base de datos interna, en comparación con el montaje y la lectura de un cartucho MVC.

En este capítulo, se describe el uso de la VLE en un entorno simple de dos sitios. No obstante, se debe tener en cuenta que la solución admite cualquier cantidad de sitios con cualquier cantidad de VLE en cada sitio. Además, uno de los sitios puede ser un sitio exclusivo de DR, donde no se ejecuten LPAR de MVS, excepto durante una prueba de DR o un desastre declarado.

El procedimiento que se indica a continuación un entorno de dos sitios: SITE1 y SITE2. Cada sitio tiene un VSM y una VLE. En este ejemplo, el SITE2 se describe como un sitio exclusivo de DR, pero el SITE2 también puede ser un sitio de producción definido como imagen reflejada de SITE1.

Nota:

El tamaño del buffer de VLE en el SITE2 debe ser suficiente para alojar tanto los datos migrados de producción como los datos creados durante una prueba de DR.

Modo de producción normal

Durante la producción normal se definen políticas en el SITE1 para migrar una copia de los datos al SITE1 de la VLE local y una segunda copia al SITE2 de la VLE remota. Se pueden crear copias adicionales, por ejemplo, ambas copias en otra VLE y copias de cinta.

A continuación, se muestra un ejemplo de políticas definidas en el SITE1:

Las definiciones de SMC se usan para asignar un nombre *MGMTCLAS* de VLEPROD a los juegos de datos con un cualificador de nivel superior de "PAYROLL".

```
POLICY NAME(VLEPOL) MEDIA(VIRTUAL) MGMT(VLEMGMT) + SUBP(VIRTSCR)
TAPEREQ DSN(PAYROLL.*) POLICY(VLEPOL)
```

Las definiciones *POOLPARAM/VOLPARAM* de HSC se usan para definir los volúmenes de producción:

```
POOLPARAM TYPE(MVC) NAME(LOCAL)
VOLPARAM VOLSER(VLL000-VLL099)
POOLPARAM TYPE(MVC) NAME(VAULT1)
VOLPARAM VOLSER(VLV000-VLV099)
POOLPARAM TYPE(SCRATCH) NAME(VIRTSCR)
VOLPARAM VOLSER(V00000-V99999) MEDIA(VIRTUAL)
```

Nota:

Tenga en cuenta que los MVC en las agrupaciones LOCAL y VAULT1 son VMVC (MVC virtuales) en el SITE1 y el SITE2 de las VLE, respectivamente, y no tienen un tipo de medio asociado a ellos.

VTCS *STORCLAS* y *MGMTCLAS* se usan para definir las políticas de VTCS:

```
STOR NAME(VLE1) STORMNGR(SITE1VLE) MVCPool(LOCAL)
STOR NAME(VLE2) STORMNGR(SITE2VLE) MVCPool(VAULT1)
MGMT NAME(VLEMGMT) DELSCR(YES) MIGPOL(VLE1,VLE2)
```

Cuando un trabajo se ejecuta con un juego de datos que comienza con el cualificador de nivel superior "PAYROLL", SMC usa *TAPEREQ* y *POLICY* para asignar una *MGMTCLAS* de VLEPROD a la solicitud de montaje. VTCS selecciona un volumen reutilizable virtual en la agrupación LOCSCR (rango V00000-V99999) y le asigna una *MGMTCLAS* de VLEPROD. Después de que el volumen se desmonta, se migra una copia a la VLE local (STORMNGR SITE1VLE) y la segunda copia se migra a la VLE remota (STORMNGR SITE2VLE).

Ejecución de una prueba de DR con la VLE

E proceso de configuración para una prueba de DR en el SITE2 es simple y rápido, y requiere restricciones mínimas en el SITE1.

Los pasos básicos son los siguientes:

1. Cree un nuevo CDS en el SITE2 que incluya solamente datos de configuración básica.
2. Marque el VMVC del SITE1 como *READONLY* para evitar conflictos.

3. Realice una auditoría de los MVC virtuales de producción en la VLE del SITE2. En este paso, se completan los CDS con los metadatos virtuales existentes. Según la cantidad de VTV en la VLE, este paso puede tardar desde unos pocos minutos hasta menos de una hora.
4. Ejecute la carga de trabajo de la prueba de DR con un rango de VTV y MVC que no se superponga con los volúmenes de producción.

En el resto de esta sección se proporcionan detalles acerca de la definición de los parámetros en el sitio de DR y se describen los pasos que se deben seguir para garantizar que el contenido de los VMVC de producción no se modifique durante la prueba.

1. Creación del CDS de prueba de DR.
 - a. Utilice el proceso *LIBGEN/SLICREAT* para crear el CDS en el SITE2. Tenga en cuenta que debe crear este CDS aunque ya esté ejecutando trabajo de producción en el SITE2. El nuevo CDS contiene solamente datos de DR del SITE1. También debe tener en cuenta que es necesario definir al menos un ACS en las macros *LIBGEN*, aunque su configuración no incluya cinta física.
 - b. Ejecute la utilidad *SET VOLPARM* para definir los volúmenes para la prueba de DR:

```
POOLPARM TYPE(MVC) NAME(VAULT1)
VOLPARM VOLSER(VLV000-VLV099)
POOLPARM TYPE(EXTERNAL) NAME(PRODVTVS)
VOLPARM VOLSER(V00000-V99999) MEDIA(VIRTUAL)
POOLPARM TYPE(MVC) NAME(DRMVC)
VOLPARM VOLSER(VLT000-VLT099)
POOLPARM TYPE(SCRATCH) NAME(VIRTSCR)
VOLPARM VOLSER(VT0000-VT9999) MEDIA(VIRTUAL)
```

Tenga en cuenta que las dos primeras agrupaciones definen volúmenes creados por el SITE1 que se utilizarán como entrada para la prueba en el SITE2. El tipo de agrupación *EXTERNAL* indica que estos son volúmenes que no forman parte de una subagrupación reutilizable. Las dos últimas agrupaciones son agrupaciones locales que se usarán como salida de la prueba en el SITE2.

- c. Defina las *MGMTCLAS* y *STORCLAS* de VTCS que se usarán para la prueba de DR:

```
STOR NAME(DRVLE) STORMNGR(SITE2VLE) MVCPOOL(DRMVC)
MGMT NAME(VLEMGMT) DELSCR(YES) MIGPOL(DRVLE)
```

- d. Tenga en cuenta que debido a que *MGMTCLAS* y las subagrupaciones reutilizables en el sistema de DR del SITE2 tienen los mismos nombres que las políticas de producción (pero distintas definiciones), ahora puede usar las mismas sentencias *POLICY* y *TAPEREQ* de SMC para la prueba de DR del SITE2 que las que usa en la producción del SITE1.
 - e. Abra el HSC/VTCS en la LPAR de prueba de DR.

2. Marque los MVC de producción como *READONLY*.
 - a. Este es un paso crítico del proceso y se debe realizar tanto en el SITE1 del CDS de producción como en el SITE2 del CDS de prueba de DR. Tenga en cuenta que una vez que los MVC se han definido como *READONLY* en el CDS de producción, puede continuar con el procesamiento normal, que incluye:

RECLAIM. La recuperación automática no selecciona un MVC en estado *READONLY* (SOLO LECTURA).

SCRATCH. Si bien los VTV se actualizarán en el CDS de producción y quedarán en estado reutilizable, es decir que se podrán reutilizar, la copia del MVC virtual de solo lectura de la VLE no se verá afectada.

Procesamiento normal para agregar o sobrescribir los VTV en los VMVC. Las nuevas versiones de VTV se migrarán a los nuevos VMVC, mientras que la copia del MVC virtual de solo lectura de la VLE no se verá afectada.

Nota:

No obstante, tenga en cuenta que no puede ejecutar la utilidad *DRAIN* en estos MVC, ya que elimina la copia de la VLE de los metadatos virtuales de MVC.

- b. Utilice la utilidad *ACTMVCGN* para seleccionar los MVC de producción en el sitio de producción mediante el CDS de producción. Esta utilidad genera sentencias de control para establecer el indicador *READONLY* en los MVC que selecciona y sentencias de control para desactivar el indicador *READONLY* una vez que la prueba ha finalizado. Al usar la palabra clave *ALL* en la sentencia de control *ACTMVCGN* se garantiza que se seleccionen todos los MVC para procesamiento *READONLY*, lo que permite que se ejecute la recuperación automática en el sistema de producción sin afectar la prueba de DR. La sentencia *SLUSMAUD DD* también se debe incluir para generar sentencias *AUDIT* para los VMVC que se usarán en la prueba. Tenga en cuenta que, si lo desea, puede ejecutar la utilidad *ACTMVCGN* en el sitio de producción para crear las actualizaciones de producción y en el sitio de DR, en una copia reflejada del CDS, para crear las actualizaciones de CDS de la prueba de DR. En el siguiente ejemplo, se muestra un JCL de muestra para ejecutar esta utilidad:

```
//ACTMVCGN JOB (ACCT), 'ACTMVCGN', NOTIFY=&SYSUID
//ACTMVCG1 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: CDS DD statements are optional if running at the production
//* site with an active HSC LPAR.
//SLSCNTL DD DSN=h1q.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=h1q.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=h1q.DBASESBY, DISP=SHR
//* NOTE: MVCMAINT READONLY(ON) STATEMENTS
//SLUMVON DD DSN=h1q.SLUMVON, DISP=(NEW, CATLG, DELETE),
```



```

//                                SPACE=(CYL,1)
/** NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUSMVOF DD DSN=h1q.SLUSMVOF,DISP=(NEW,CATLG,DELETE),
//                                SPACE=(CYL,1)
/**      NOTE: AUDIT MVC(VVVVVV) STATEMENTS
//SLUSMAUD DD DSN=h1q.SLUSMAUD,DISP=(NEW,CATLG,DELETE),
//                                SPACE=(CYL,1)
/** NOTE: THE FOLLOWING SELECTS ALL "NON-EMPTY" VMVCS
//SLSIN DD *
ACTMVCN ALL MVCPOOL(VAULT1)
/*

```

3. En el sitio de producción, ejecute la función de la utilidad *MVCMAINT* para marcar los VMVC como *READONLY*.

```

//RONLYON EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
/** NOTE: EXEC MVCMAINT TO SET READONLY(ON). Output of
/**      ACTMVCN utility.
//SLSIN DD DSN=h1q.SLUSMVON,DISP=SHR

```

4. Abra el HSC/VTCS en el sitio de DR.
5. Ejecute una autoría de MVC de los VMVC de producción en el SITE2 de la VLE con el CDS del SITE2 recientemente creado y la salida de la utilidad *ACTMVCN*. En este paso, se completan los metadatos de CDS que contienen las relaciones entre los VTV y los VMVC.

```

//AUDIT EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
/**      NOTE: AUDIT CONTROL STATEMENTS FROM ACTMVCN UTILITY
//SLSIN DD DSN=h1q.SLUSMAUD,DISP=SHR

```

De manera opcional, puede recuperar los VTV que se utilizarán en la prueba de DR en el buffer de VTSS, mediante LCM u otro método de selección de VTV para recuperación. No obstante, este paso no es necesario, ya que las recuperaciones a partir del buffer de la VLE son relativamente rápidas.

6. Ejecute la utilidad *MVCMAINT* mediante la salida de *ACTMVCN READONLY(ON)* para establecer los VMVC de producción en *READONLY* en el SITE2, en el CDS de DR.

```

//RONLYON EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*

```

```

/* NOTE: EXEC MVCMAINT TO SET READONLY(ON). Output of
/*      ACTMVCGN utility.
//SLSIN      DD DSN=h1q.SLUSMVON,DISP=SHR

```

7. Opcional: Antes de comenzar la prueba de DR, puede ejecutar *VTVRPT* y *MVCRPT* para validar el contenido del CDS de prueba de DR.
8. Ejecute la carga de trabajo de prueba de DR.
 - a. Abra SMC. Si usó los mismos nombres en la *MGMTCLAS* y las subagrupaciones reutilizables que en el sistema de producción, puede usar sus sentencias de producción *TAPEREQ* y *POLICY*. Se recomienda, pero no es obligatorio, que utilice un nombre de TapePlex distinto para el TapePlex de prueba de DR.
 - b. Ejecute la carga de trabajo de DR con el SMC y el nuevo CDS de HSC/VTCS.
 - c. No hay limitaciones sobre la actualización de volúmenes de producción de VTV durante las pruebas de DR. Los datos de los VTV de producción se pueden agregar a (*DISP=MOD*) o sobrescribir (*DISP=OLD*). Estas actualizaciones no afectan el contenido de la copia de VTV en el MVC de producción virtual *READONLY* y, por lo tanto, no afectan la copia de producción de los datos.

Limpeza después de una prueba de DR con VLE

Una vez que la prueba de DR ha finalizado, el objetivo de la limpieza es eliminar metadatos del VTSS y de la VLE, de modo que la siguiente prueba de DR no vea estos datos. Tenga en cuenta que el HSC/VTCS de la prueba de DR debe permanecer activo hasta que la limpieza haya finalizado. Los pasos son los siguientes:

1. Ejecute la función de la utilidad *SCRATCH* para reutilizar todos los VTV creados durante la prueba del VTSS y de los VMVC de la prueba de DR de la VLE. Cuando el parámetro *DELSCR(YES)* se especifica en la prueba de DR *MGMTCLAS*, la ejecución de la utilidad de reutilización hace que los VTV se supriman de los metadatos de la VLE y del buffer.

```

//SCRATCH EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
SCRATCH VOL(VT0000-VT9999)

```

Tenga en cuenta que si modificó cualquier VTV de producción mediante *DISP=MOD* o *DISP=OLD*, estos VTV permanecen en el buffer y en la VLE.

Al reutilizar los VTV de la subagrupación de la prueba DR después de la prueba, minimiza la cantidad de tiempo que se necesita para limpiar el VTSS y la cantidad de datos que quedan en la VLE después de la finalización de la prueba.

2. Migre el VTSS a 0.

```
//MIGRTO0 EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
MIGRATE VTSS(DRVTSS) THRESHLD(0)
```

Este paso se debe realizar solamente si la salida de su prueba de DR incluía nuevas versiones de VTV de producción.

3. Verifique que el VTSS de DR ahora esté vacío.

```
//AUDVTSS EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
AUDIT VTSS(DRVTSS)
```

Tenga en cuenta que si modificó los VTV de producción durante la prueba de DR, las copias de estos datos y los metadatos permanecen en la VLE para la agrupación de MVC de la prueba de DR (VLT000-VLT099, VTV V00000-V99999). Durante la siguiente prueba de DR, estos VMVC se escribirán a partir del comienzo lógico de la cinta, y cualquier dato que contengan se eliminará de la VLE. Dado que el nuevo CDS de prueba de DR no tiene conocimiento de estos datos, no afectará a la siguiente prueba de DR.

4. En el sitio de producción, use las tarjetas de control *READONLY(OFF)* creadas por la utilidad *ACTMVCGN* al comienzo de la prueba para colocar los VMVC de producción nuevamente en un estado que admita escritura.

```
//RDONLYOF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: EXEC MVCMAINT TO SET READONLY(OFF)
//SLSIN DD DSN=h1q.SLUSMVOF,DISP=SHR
```

Uso de la VLE para continuidad empresarial

Cuando se produce una interrupción en el SITE1 que requiere que el SITE2 tome la carga de trabajo del SITE1, el proceso es casi idéntico al procedimiento de prueba de DR.

Si se está ejecutando una prueba de DR cuando se produce la interrupción del SITE1, siga el proceso que se describe anteriormente para realizar una limpieza después de la prueba de DR y detenga la prueba de DR.

Para comenzar a ejecutar la carga de trabajo del SITE1 en el SITE2, siga el procedimiento descrito anteriormente para iniciar una prueba de DR. Obviamente, omitirá el paso que

consiste en marcar los VMVC de producción como *READONLY* en el CDS de producción, ya que no hay un CDS de "producción" para actualizar. No obstante, utilizará la copia reflejada del CDS de producción para generar las tarjetas de control *MVCMAINT READONLY* para los MVC de producción en la VLE.

También tendrá que usar las políticas de prueba de DR que separan los VTV que se están creando y los VMVC de salida en un rango separado, a fin de evitar cualquier posibilidad de dañar los datos de producción hasta después de que se haya verificado la continuidad empresarial.

Nota:

Si los trabajos de producción que realizan el procesamiento de *DISP=MOD* para datos de cinta no tienen un punto de sincronización definido, puede que el contenido de un VTV en el momento de una interrupción sea impredecible. StorageTek recomienda revisar todos los procedimientos de recuperación ante desastres a fin de garantizar puntos de sincronización predecibles para los datos de cinta.

Apéndice A. Ejemplos de VTSS en cluster

En "*Uso de configuraciones de VTSS en cluster*", se describen los aspectos básicos de la agrupación en clusters del VTSS y, en este apéndice, se suministran los siguientes ejemplos de agrupaciones en clusters:

- "VTSS en cluster unidireccional"
- "VTSS en cluster bidireccional"
- "Agrupación en clusters ampliada"
- "Cluster VSM5 a VSM5 con CLINK TCP/IP"
- "Cluster VSM5 a VSM 6 con CLINK TCP/IP y VLE interconectadas"
- "Cluster VSM 6 a VSM 6 "sin cinta" con CLINK TCP/IP"
- "¿Debe usar una configuración unidireccional o bidireccional?"

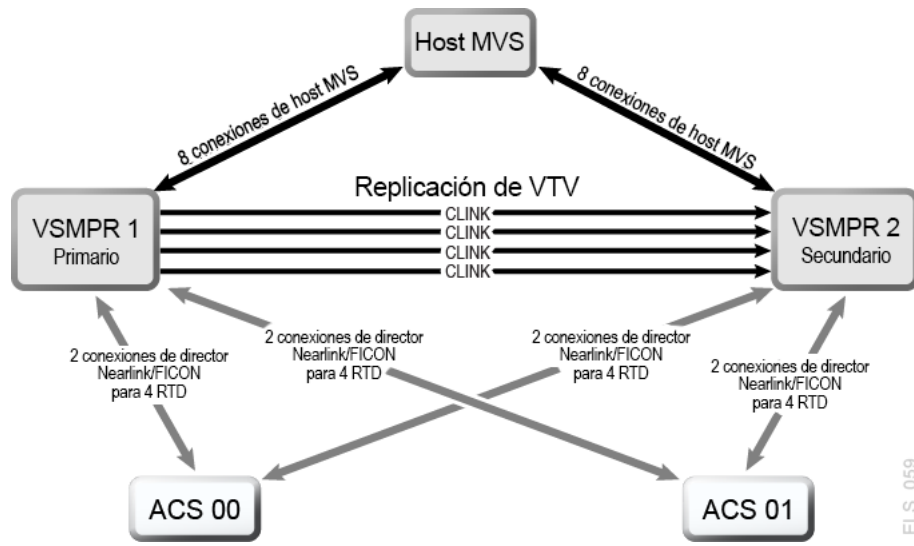
VTSS en cluster unidireccional

En *Figura A.1, "Configuración de VTSS en cluster unidireccional con dos ACS"*, se muestra un ejemplo de un sistema de VTSS en cluster unidireccional con dos ACS. Tenga en cuenta que en este ejemplo, los puertos FICON proporcionan las conexiones CLINK. En este ejemplo, hay un solo host de MVS, pero está generando una gran cantidad de datos críticos que debe proteger con dos nuevos VSM4 que acaba de comprar.

VTSS1 es el VTSS principal y está conectado al secundario (VTSS2) mediante enlaces de cluster (CLINK). Si la clase de gestión para un VTV especifica replicación, cuando el VTV llegue al VTSS1, se replicará (copiará) en el VTSS2 e inmediatamente se migrará (mediante KEEP).

Como consecuencia, ha aumentado la disponibilidad de los datos (hay una copia del VTV en cada VTSS, en caso de que uno falle) y la protección de los datos (el VTV también está en cintas magnéticas en carcasas cuadradas, en caso de que ambos VTSS queden fuera de línea). Por lo tanto, el VTSS en cluster es una gran solución, tanto para la continuidad empresarial como para la reanudación de las actividades empresariales.

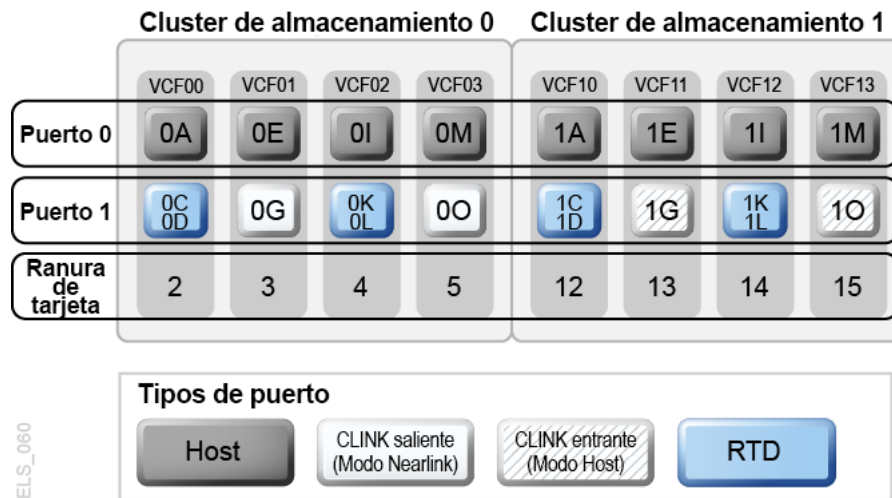
Figura A.1. Configuración de VTSS en cluster unidireccional con dos ACS



Ahora, observe el hardware de esta configuración en cluster. En [Figura A.2, “VSM4 con 8 tarjetas VCF, 8 puertos de host, directores FICON para 8 RTD y 4 puertos de CLINK”](#), se muestran identificadores CONFIG de interfaz de canal para un VSM4 con 8 tarjetas VCF. En esta configuración, asignó:

- 8 puertos de host
- 4 puertos para RTD. Los puertos de RTD están conectados a directores FICON, cada uno de los cuales está conectado a las RTD; por lo tanto, los identificadores *CHANIF* de ambas RTD se muestran en cada puerto. Esto permite conexiones backend a 8 RTD, aunque solo una RTD por puerto/director puede estar activa a la vez.
- 4 puertos para que las conexiones CLINK formen un cluster de VTSS unidireccional y 8 puertos para conexiones host. Para formar el VTSS en cluster, hay dos VSM4 (VTSS1 y VTSS2) configurados de manera idéntica, como se muestra en [Figura A.2, “VSM4 con 8 tarjetas VCF, 8 puertos de host, directores FICON para 8 RTD y 4 puertos de CLINK”](#).

Figura A.2. VSM4 con 8 tarjetas VCF, 8 puertos de host, directores FICON para 8 RTD y 4 puertos de CLINK



Ahora, ha observado qué aspecto tiene un ejemplo de cluster unidireccional y cuáles son las configuraciones de tarjeta VCF necesarias. Ahora, diríjase a "[Configuración y gestión de un sistema de VTSS en cluster unidireccional.](#)"

Configuración y gestión de un sistema de VTSS en cluster unidireccional

Para configurar y gestionar un sistema en cluster unidireccional, como se muestra en [Figura 6.1, "VTSS en cluster unidireccional"](#), haga lo siguiente:

1. Asegúrese de que su sistema cumpla con los requisitos de VTSS en cluster.
2. Use *CONFIG* para crear las sentencias *CLUSTER* y *CLINK* para definir el cluster de VTSS y sus conexiones.

En el siguiente ejemplo, se muestra *CONFIG JCL* para definir un cluster unidireccional de dos VSM4 (VTSS1 y VTSS2), como se muestra en [Figura 6.1, "VTSS en cluster unidireccional"](#).

Tenga en cuenta lo siguiente:

- La sentencia *CLUSTER* define el cluster como un cluster que está formado por VTSS1 y VTSS2.
- Hay sentencias *CLINK* que usan los puertos emisores (modo Nearlink) de **VTSS1 solamente** para activar el cluster como unidireccional, donde VTSS1 es el principal y VTSS2, el secundario.

```
//CREATECFG EXEC PGM=SWSADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SLSLINK,DISP=SHR
```

```

//SLS_CNTL DD DSN=FEDB.VSMLMULT.DBASEPRM,DISP=SHR
//SLS_CNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASETBY,DISP=SHR
//CFG22202 DD DSN=FEDB.VSMLMULT.CFG22202,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
VTVVOL LOW=905000 HIGH=999999 SCRATCH
VTVVOL LOW=C00000 HIGH=C25000 SCRATCH
VTVVOL LOW=RMM000 HIGH=RMM020 SCRATCH
MVCVOL LOW=N25980 HIGH=N25989
MVCVOL LOW=N35000 HIGH=N35999
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PR11A00 DEVNO=1A00 CHANIF=0C
RTD NAME=PR11A01 DEVNO=1A01 CHANIF=0D
RTD NAME=PR11A02 DEVNO=1A02 CHANIF=0K
RTD NAME=PR11A03 DEVNO=1A03 CHANIF=0L
RTD NAME=PR12A08 DEVNO=2A08 CHANIF=1C
RTD NAME=PR12A09 DEVNO=2A09 CHANIF=1D
RTD NAME=PR12A0A DEVNO=2A0A CHANIF=1K
RTD NAME=PR12A0B DEVNO=2A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
VTSS NAME=VTSS2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PR23A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PR23A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PR23A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PR23A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PR24A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PR24A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PR24A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PR24A0B DEVNO=4A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSSs(VTSS1,VTSS2)
CLINK VTSS=VTSS1 CHANIF=0G
CLINK VTSS=VTSS1 CHANIF=00
CLINK VTSS=VTSS1 CHANIF=1G
CLINK VTSS=VTSS1 CHANIF=10

```

3. Especifique la configuración de replicación condicional en la sentencia *CONFIG GLOBAL*.

```
CONFIG GLOBAL REPLICAT=CHANGED
```


En este ejemplo, *CONFIG GLOBAL REPLICAT=CHANGED* especifica:

- Replicar los VTV solo si el VTV está actualizado y no existe una copia idéntica en el secundario.
- Mediante el parámetro *MIGPOL*, migre los VTV duplicados a los ACS 00 y 01, por las clases de almacenamiento que creó en el paso 5.

Para replicar de manera incondicional los VTV, especifique *CONFIG GLOBAL REPLICAT=ALWAYS*.

4. Cree una clase de gestión que especifique la replicación de VTV y dos clases de almacenamiento para migrar (duplicados) los VTV replicados.

```
MGMT NAME(VSMREPL) REPLICAT(YES) MIGPOL(REPLSTR1,REPLSTR2)
```

Nota:

- Tenga en cuenta la interacción entre *GLOBAL REPLICAT*, que especifica cuándo puede ocurrir la replicación, y *MGMTClas REPLICAT(YES)*, que indica que cuando la condición *GLOBAL REPLICAT* dice que es el momento, se debe realizar la replicación.
- La clase de gestión *VSMREPL* no especifica una política de migración inmediata. La replicación de VTV aplica la migración de inmediato. Los VTV de esta clase de gestión se agregan a la cola de migración inmediata en el VTSS, una vez que la replicación ha finalizado. Tenga en cuenta que la duplicación no es un requisito para la replicación de los VTV. Para obtener más información, consulte "[Cómo funcionan las configuraciones de VTSS en cluster.](#)"

5. Cree las clases de almacenamiento para los MVC que contienen los VTV replicados y migrados.

```
STOR NAME(REPLSTR1) ACS(00) MEDIA(STK1R) MIRATE(RECEIVER)
STOR NAME(REPLSTR2) ACS(01) MEDIA(STK1R) MIGRATE(RECEIVER)
```

En este ejemplo, la sentencia *STORClas* define las clases de almacenamiento *REPLSTR1* y *REPLSTR2* a las que se hace referencia en el parámetro *MIGPOL*, en el paso 4. También debe tener en cuenta que los parámetros *MIGRATE* de la clase de almacenamiento especifican que el VTSS que recibe los VTV replicados, en este caso, VTSS2 (secundario), hace la migración a ambos ACS. Esta es una manera de asegurarse de que el secundario funcione como el motor de la migración.

6. Cargue las sentencias de control *MGMTClas* y *STORClas* con el comando *MGMTDEF*.

```
MGMTDEF DSN(hsc.parms)
```

7. Cree una sentencia *TAPEREQ* para enrutar los datos críticos a VSM y asignar la clase de gestión *VSMREPL* a los datos.

```
TAPEREQ DSN(*.PAYROLL.***) MEDIA(VIRTUAL) MGMT(VSMREPL)
```

En este ejemplo, la sentencia *TAPEREQ* especifica:

- Enrutar juegos de datos con la máscara de HLQ **.PAYROLL.*** a VSM.
- Asignar la clase de gestión VSMREPL que creó en el paso 4.

Atención:

Para replicar VTV, tanto VTSS1 como VTSS2 se deben cambiar al estado en línea para VTCS, de modo que VTCS pueda enviar comandos de control a ambos VTSS. Consulte "[Cómo funcionan las configuraciones de VTSS en cluster](#)" para obtener más información.

También puede usar la sustitución de grupos esotéricos mediante la sentencia *TAPEREQ* de SMC o rutinas DFSMS del ACS de SMC para enrutar trabajos de replicación a VSM. Para obtener más información, consulte la *Guía de Administración y Configuración de SMC*.

8. Compruebe las opciones de PARMLIB del HSC para asegurarse de que los registros del subtipo 28 estén activados.

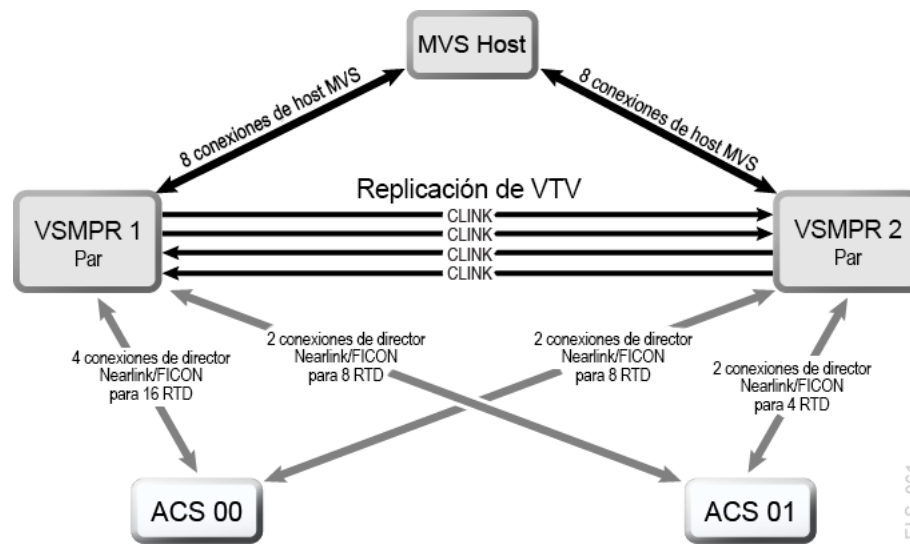
Si están activados, la agrupación en clusters del VTSS escribe un registro del subtipo 28 por cada replicación realizada.

VTSS en cluster bidireccional

En [Figura A.3, “Configuración de VTSS en cluster bidireccional con dos ACS”](#), se muestra un ejemplo de un sistema de VTSS en cluster bidireccional con dos ACS. Tenga en cuenta que en este ejemplo, los puertos FICON proporcionan las conexiones CLINK.

Este sistema es muy similar al del ejemplo del sistema unidireccional, pero va un paso más allá: hay dos hosts de MVS que comparten un CDS y todo lo que se observa en la imagen está interconectado. Estos sitios se reflejan entre sí para lograr la mejor disponibilidad de los datos y la mejor protección posibles. Para transformarlo en un sistema bidireccional, debe configurar los dos VTSS como pares mediante las sentencias *CLINK*.

Figura A.3. Configuración de VTSS en cluster bidireccional con dos ACS

**Nota:**

- La agrupación en clusters bidireccionales requiere VTCS 6.1 y superiores. No puede configurar un cluster bidireccional con las versiones inferiores a VTCS 6.1.
- Esta configuración se muestra con la función que permite un total de hasta 16 transferencias simultáneas de E/S en modo NearLink. Se pueden distribuir en varios destinos, en 14 puertos NearLink como máximo y se pueden realizar hasta dos transferencias simultáneas de E/S en modo NearLink por puerto. Esta función requiere microcódigo D02.06.00.00 de VTSS o superior.

En Figura A.4, “VSMPR1: VSM5 con 8 tarjetas VCF, 8 puertos de host, directores FICON para 24 RTD, 4 CLINK”, se muestran los identificadores *CONFIG* de interfaz de canal de VSMPR1 que se pueden ver en Figura A.3, “Configuración de VTSS en cluster bidireccional con dos ACS”. En esta configuración, asignó:

- 8 puertos de host
- 6 puertos para RTD. Los puertos de RTD están conectados a directores FICON, cada uno de los cuales está conectado a 4 RTD; por lo tanto, los identificadores CHANIF de las 4 RTS se muestran en cada puerto. Esto permite conexiones backend a 24 RTD para ACS00, aunque solo una RTD por puerto/director puede estar activa a la vez.
- 4 puertos que usan directores FICON. Dos son puertos Nearlink para el creador, dos están en el modo host para el terminador para que las conexiones CLINK formen un cluster de VTSS bidireccionales.

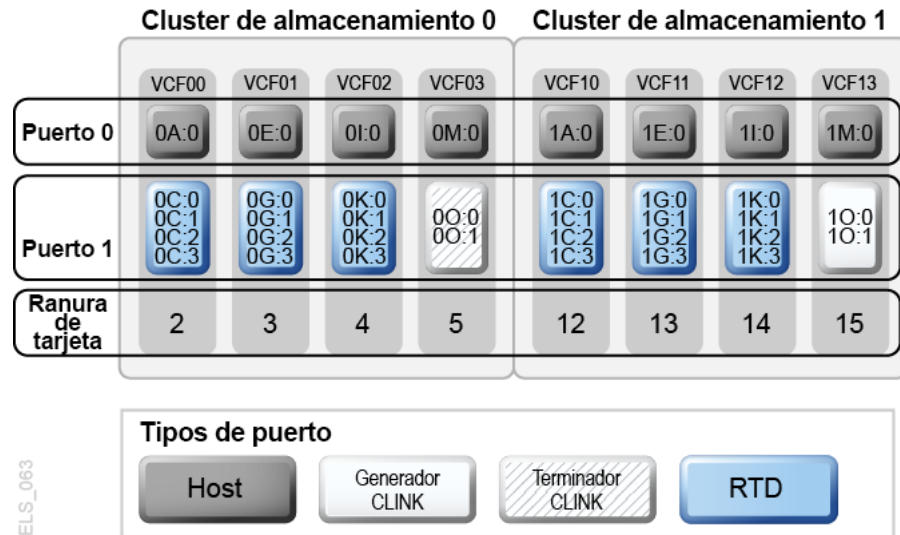
Figura A.4. VSMRP1: VSM5 con 8 tarjetas VCF, 8 puertos de host, directores FICON para 24 RTD, 4 CLINK



En [Figura A.5, “VSMRP2: VSM5 con 8 tarjetas VCF, 8 puertos de host, directores FICON para 24 RTD, 4 CLINK”](#), se muestran los identificadores *CONFIG* de interfaz de canal para un VSMRP1, un VSM5 en un cluster bidireccional con 8 tarjetas VCF y la función de 32 RTD como máximo activada. En esta configuración, asignó:

- 8 puertos de host
- 6 puertos para RTD. Los puertos de RTD están conectados a directores FICON, cada uno de los cuales está conectado a 4 RTD; por lo tanto, los identificadores *CHANIF* de las 4 RTD se muestran en cada puerto. Esto permite conexiones backend a 24 RTD, aunque solo una RTD por puerto/director puede estar activa a la vez.

Figura A.5. VSMR2: VSM5 con 8 tarjetas VCF, 8 puertos de host, directores FICON para 24 RTD, 4 CLINK



- 4 puertos que usan directores FICON. Dos son puertos Nearlink para el creador, dos están en el modo host para el terminador para que las conexiones CLINK formen un cluster de VTSS bidireccionales.

Atención:

Como se muestra en [Figura 6.3, “CLINK ESCON/FICON para VTSS en cluster bidireccionales”](#), es un requisito que cada CLINK debe estar conectado al mismo cluster de almacenamiento en cada VTSS. Si no puede realizar la configuración de esta manera, se pueden producir errores de replicación, canal y comunicación. Por lo tanto, como se muestra, los puertos Nearlink (creadores de CLINK) en VSMR1 están en el cluster de almacenamiento 0 y los puertos de host (terminadores de CLINK) en VSMR2 también están en el cluster de almacenamiento 0. Lo mismo se aplica a las conexiones CLINK para los datos que fluyen en la dirección opuesta; ambos están en el cluster de almacenamiento 1.

Configuración y gestión de un sistema de VTSS en cluster bidireccional

Para configurar y gestionar un sistema en cluster bidireccional, como se muestra en [Figura A.3, “Configuración de VTSS en cluster bidireccional con dos ACS”](#), haga lo siguiente:

1. Asegúrese de que su sistema cumpla con los requisitos de VTSS en cluster descritos en *Instalación de ELS*.
2. Use *CONFIG* para crear las sentencias *CLUSTER* y *CLINK* para definir el cluster de VTSS y sus conexiones.

En el siguiente ejemplo, se muestra CONFIG JCL para definir un cluster bidireccional de dos VSM4s (VSMPR1 y VSMPR2), como se muestra en [Figura A.3, “Configuración de VTSS en cluster bidireccional con dos ACS”](#).

- Tenga en cuenta que la sentencia *CLUSTER* define el cluster como un cluster formado por VSMPR1 y VSMPR2.
- Hay sentencias *CLINK* que usan los puertos emisores (modo Nearlink) de ambos VTSS para activar el cluster como bidireccional y se conectan con el mismo cluster de almacenamiento en cada VTSS para los puertos emisores y receptores de cada CLINK.

```
//CREATECF EXEC PGM=SWSADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SLSLINK,DISP=SHR
//SLSCNTL DD DSN=FEDB.VSMLMULT.DBASEPRM,DISP=SHR
//SLSCNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASEBY,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MCVFREE=40 VTVATTR=SCRATCH RECALWER=YES LOCKSTR=VTCS_LOCKS
REPLICAT=ALWAYS VTVPAGE=LARGE SYNCHREP=YES MAXRTDS=32
RECLAIMTHRESHLD=70 MAXMVC=40 START=35
RECLAIMTHRESHLD=70 MAXMVC=40 START=35
VTVVOL LOW=905000 HIGH=999999 SCRATCH
VTVVOL LOW=C00000 HIGH=C25000 SCRATCH
VTVVOL LOW=RMM000 HIGH=RMM020 SCRATCH
MVCVOL LOW=N25980 HIGH=N25989
MVCVOL LOW=N35000 HIGH=N35999
VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=VPR12A00 DEVNO=2A00 CHANIF=0C:0
RTD NAME=VPR12A01 DEVNO=2A01 CHANIF=0C:1
RTD NAME=VPR12A02 DEVNO=2A02 CHANIF=0C:2
RTD NAME=VPR12A03 DEVNO=2A03 CHANIF=0C:3
RTD NAME=VPR12A04 DEVNO=2A04 CHANIF=0G:0
RTD NAME=VPR12A05 DEVNO=2A05 CHANIF=0G:1
RTD NAME=VPR12A06 DEVNO=2A06 CHANIF=0G:2
RTD NAME=VPR12A07 DEVNO=2A07 CHANIF=0G:3
RTD NAME=VPR12A08 DEVNO=2A08 CHANIF=0K:0
RTD NAME=VPR12A09 DEVNO=2A09 CHANIF=0K:1
RTD NAME=VPR12A0A DEVNO=2A0A CHANIF=0K:2
RTD NAME=VPR12A0B DEVNO=2A0B CHANIF=0K:3
RTD NAME=VPR13A00 DEVNO=3A00 CHANIF=1C:0
RTD NAME=VPR13A01 DEVNO=3A01 CHANIF=1C:1
RTD NAME=VPR13A02 DEVNO=3A02 CHANIF=1C:2
RTD NAME=VPR13A03 DEVNO=3A03 CHANIF=1C:3
RTD NAME=VPR13A04 DEVNO=3A04 CHANIF=1G:0
```

```

RTD  NAME=VPR13A05  DEVNO=3A05  CHANIF=1G:1
RTD  NAME=VPR13A06  DEVNO=3A06  CHANIF=1G:2
RTD  NAME=VPR13A07  DEVNO=3A07  CHANIF=1G:3
RTD  NAME=VPR13A08  DEVNO=3A08  CHANIF=1K:0
RTD  NAME=VPR13A09  DEVNO=3A09  CHANIF=1K:1
RTD  NAME=VPR13A0A  DEVNO=3A0A  CHANIF=1K:2
RTD  NAME=VPR13A0B  DEVNO=3A0B  CHANIF=1K:3
VTD  LOW=9900  HIGH=99FF
VTSS NAME=VSMR2  LOW=70  HIGH=80  MAXMIG=8  MINMIG=4  RETAIN=5
RTD  NAME=VPR22B00  DEVNO=2B00  CHANIF=0C:0
RTD  NAME=VPR22B01  DEVNO=2B01  CHANIF=0C:1
RTD  NAME=VPR22B02  DEVNO=2B02  CHANIF=0C:2
RTD  NAME=VPR22B03  DEVNO=2B03  CHANIF=0C:3
RTD  NAME=VPR22B04  DEVNO=2B04  CHANIF=0G:0
RTD  NAME=VPR22B05  DEVNO=2B05  CHANIF=0G:1
RTD  NAME=VPR22B06  DEVNO=2B06  CHANIF=0G:2
RTD  NAME=VPR22B07  DEVNO=2B07  CHANIF=0G:3
RTD  NAME=VPR22B08  DEVNO=2B08  CHANIF=0K:0
RTD  NAME=VPR22B09  DEVNO=2B09  CHANIF=0K:1
RTD  NAME=VPR22B0A  DEVNO=2B0A  CHANIF=0K:2
RTD  NAME=VPR22B0B  DEVNO=2B0B  CHANIF=0K:3
RTD  NAME=VPR23B00  DEVNO=3B00  CHANIF=1C:0
RTD  NAME=VPR23B01  DEVNO=3B01  CHANIF=1C:1
RTD  NAME=VPR23B02  DEVNO=3B02  CHANIF=1C:2
RTD  NAME=VPR23B03  DEVNO=3B03  CHANIF=1C:3
RTD  NAME=VPR23B04  DEVNO=3B04  CHANIF=1G:0
RTD  NAME=VPR23B05  DEVNO=3B05  CHANIF=1G:1
RTD  NAME=VPR23B06  DEVNO=3B06  CHANIF=1G:2
RTD  NAME=VPR23B07  DEVNO=3B07  CHANIF=1G:3
RTD  NAME=VPR23B08  DEVNO=3B08  CHANIF=1K:0
RTD  NAME=VPR23B09  DEVNO=3B09  CHANIF=1K:1
RTD  NAME=VPR23B0A  DEVNO=3B0A  CHANIF=1K:2
RTD  NAME=VPR23B0B  DEVNO=3B0B  CHANIF=1K:3
VTD  LOW=9900  HIGH=99FF
CLUSTER NAME=CLUSTER1  VTSSs(VSMR1,VSMR2)
CLINK VTSS=VSMR1  CHANIF=00:0
CLINK VTSS=VSMR1  CHANIF=00:1
CLINK VTSS=VSMR2  CHANIF=10:0
CLINK VTSS=VSMR2  CHANIF=10:1

```

3. Especifique la configuración de replicación condicional en la sentencia *CONFIG GLOBAL*.

```
CONFIG GLOBAL REPLICAT=CHANGED
```

Al igual que en el ejemplo del sistema unidireccional, en este ejemplo, use *CONFIG GLOBAL REPLICAT=CHANGED*.

4. Cree una clase de gestión que especifique la replicación de VTV y dos clases de almacenamiento para migrar (duplicados) los VTV replicados.

```
MGMT NAME(VSMREPL) REPLICAT(YES) MIGPOL(REPLSTR1,REPLSTR2)
```

En este ejemplo, replique los VTV solo si cambiaron, pero no en el otro VTSS del cluster. Migre los duplicados a los ACS 01 y 00 por las clases de almacenamiento que creará en el paso 5.

5. Cree las clases de almacenamiento para los MVC que contienen los VTV replicados y migrados.

```
STOR NAME(REPLSTR1) ACS(01) MEDIA(STK1R) MIRATE(EITHER)
STOR NAME(REPLSTR2) ACS(00) MEDIA(STK1R) MIGRATE(EITHER)
```

En este ejemplo, la sentencia *STORcLas* define las clases de almacenamiento *REPLSTR1* y *REPLSTR2* a las que se hace referencia en el parámetro *MIGPOL*, en el paso 4. También debe tener en cuenta que para optimizar los recursos de VTSS y RTD, los parámetros *MIGRATE* de las clases de almacenamiento permiten migraciones desde cualquier VTSS. Esta es una estrategia habitual para clusters de VTSS bidireccionales de par a par.

6. Cargue las sentencias de control *MGMTcLas* y *STORcLas* con el comando *MGMTDEF*.

```
MGMTDEF DSN(hsc.parms)
```

7. Cree una sentencia *TAPEREQ* para enrutar los datos críticos a VSM y asignar la clase de gestión *VSMREPL* a los datos.

```
TAPEREQ DSN(*.PAYROLL.***) MEDIA(VIRTUAL) MGMT(VSMREPL)
```

En este ejemplo, la sentencia *TAPEREQ* especifica:

- Enrutar juegos de datos con la máscara de HLQ **.PAYROLL.**** a VSM.
- Asignar la clase de gestión *VSMREPL* que activó en el paso 4.

Atención:

Para replicar VTV, tanto *VSMR1* como *VSMR2* se deben cambiar al estado en línea para VTCS, de modo que VTCS pueda enviar comandos de control a ambos VTSS. Consulte "[Cómo funcionan las configuraciones de VTSS en cluster](#)" para obtener más información.

- También puede usar la sustitución de grupos esotéricos mediante la sentencia *TAPEREQ* de SMC o salidas de usuario de ELS para enrutar trabajos de replicación a VSM. Si se

sustituye un grupo esotérico que distribuye todos los VTD en todos los VTSS pares, el VTCS puede seguir influyendo en la asignación correctamente si uno de los VTSS pares de un cluster se deja fuera de línea.

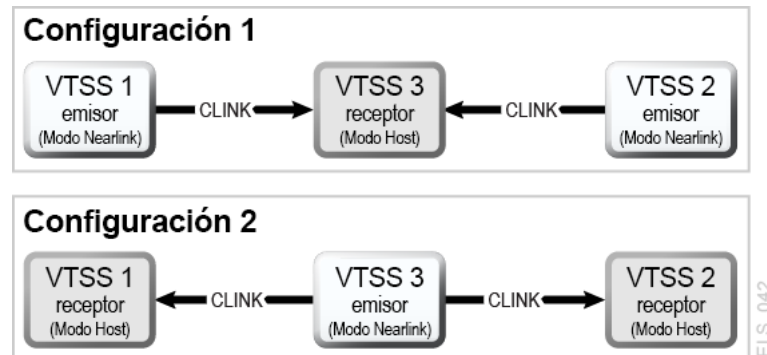
- Si se asigna un nombre de clase de gestión a SMC en la interfaz DFSMS de StorageTek, estará disponible en el momento de la asignación. Por lo tanto, no será necesario que el grupo esotérico asignado en la interfaz incluya solo los VTSS que forman parte de los clusters. Siempre que un grupo esotérico contenga algunas unidades asignadas en el VTSS principal de un cluster de funcionamiento completo, el SMC tendrá suficiente información para dirigir la asignación a una unidad de un VTSS principal si la clase de gestión específica que la replicación está activada.
8. Compruebe las opciones de PARMLIB del HSC para asegurarse de que los registros del subtipo 28 estén activados.

Si están activados, la agrupación en clusters del VTSS escribe un registro del subtipo 28 por cada replicación realizada.

Agrupación en clusters ampliada

La agrupación en clusters ampliada (EC) permite la conexión de tres o más VTSS mediante CLINK con una sola configuración de TapePlex (1 CDS), como se muestra en el ejemplo de [Figura A.6, “Configuraciones básicas de una agrupación en clusters ampliada”](#).

Figura A.6. Configuraciones básicas de una agrupación en clusters ampliada



Configuración y gestión de un sistema de 3 VTSS en cluster

Como se muestra en [Figura A.6, “Configuraciones básicas de una agrupación en clusters ampliada”](#), la configuración 1 muestra 2 VTSS replicándose a un solo VTSS "recopilador", que es la configuración más práctica, porque una ubicación principal con varios VSM puede alimentar a los VTV de una ubicación secundaria con un solo VSM "recopilador". La replicación sincrónica y la asincrónica están disponibles para su uso en cada VTSS emisor. Cada VTSS debe tener RTS equivalentes (modelo similar) conectadas. Como se muestra en las sentencias CONFIG para la configuración 1 que aparece a continuación:

- La sentencia *CLUSTER* define todos los nombres de VTSS configurados para agrupación en clusters.
- La sentencia *CLINK* define la ubicación del puerto Nearlink en el VTSS emisor y su PAR o VTSS de destino.

```
/CREATCFG EXEC PGM=SLUADMIN, PARM='MIXED'  
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR  
//SLS_CNTL DD DSN=h1q.DBASEPRM, DISP=SHR  
//SLS_CNTL2 DD DSN=h1q.DBASESEC, DISP=SHR  
//SLSSTBY DD DSN=h1q.DBASETBY, DISP=SHR  
//SLSPRINT DD SYSOUT=*  
//SLSIN DD *  
CONFIG RESET CDSLEVEL(V62ABOVE)  
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES  
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE  
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5  
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5  
RTD NAME=PA11A00 DEVNO=1A00 CHANIF=0C  
RTD NAME=PA11A01 DEVNO=1A01 CHANIF=0D  
RTD NAME=PA11A02 DEVNO=1A02 CHANIF=0K  
RTD NAME=PA11A03 DEVNO=1A03 CHANIF=0L  
RTD NAME=PA12A08 DEVNO=2A08 CHANIF=1C  
RTD NAME=PA12A09 DEVNO=2A09 CHANIF=1D  
RTD NAME=PA12A0A DEVNO=2A0A CHANIF=1K  
RTD NAME=PA12A0B DEVNO=2A0B CHANIF=1L  
VTD LOW=7900 HIGH=79FF  
VTSS NAME=VTSS2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5  
RTD NAME=PA23A00 DEVNO=3A00 CHANIF=0C  
RTD NAME=PA23A01 DEVNO=3A01 CHANIF=0D  
RTD NAME=PA23A02 DEVNO=3A02 CHANIF=0K  
RTD NAME=PA23A03 DEVNO=3A03 CHANIF=0L  
RTD NAME=PA24A08 DEVNO=4A08 CHANIF=1C  
RTD NAME=PA24A09 DEVNO=4A09 CHANIF=1D  
RTD NAME=PA24A0A DEVNO=4A0A CHANIF=1K  
RTD NAME=PA24A0B DEVNO=4A0B CHANIF=1L  
VTD LOW=8900 HIGH=89FF  
VTSS NAME=VTSS3 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5  
RTD NAME=PA33A00 DEVNO=3A00 CHANIF=0C  
RTD NAME=PA33A01 DEVNO=3A01 CHANIF=0D  
RTD NAME=PA33A02 DEVNO=3A02 CHANIF=0K  
RTD NAME=PA33A03 DEVNO=3A03 CHANIF=0L  
RTD NAME=PA34A08 DEVNO=4A08 CHANIF=1C  
RTD NAME=PA34A09 DEVNO=4A09 CHANIF=1D  
RTD NAME=PA34A0A DEVNO=4A0A CHANIF=1K  
RTD NAME=PA34A0B DEVNO=4A0B CHANIF=1L
```

```
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSS(VTSS1,VTSS2,VTSS3)
CLINK VTSS=VTSS1 CHANIF=0G PART=VTSS3
CLINK VTSS=VTSS2 CHANIF=0G PART=VTSS3
```

Como se muestra en [Figura A.6, “Configuraciones básicas de una agrupación en clusters ampliada”](#), la configuración 2 muestra un solo VTSS en replicación conectado a dos VTSS receptores. Tenga en cuenta que el término "recopilador" no se utilizó aquí, porque un VTV solo se replica a un VTSS, ya sea VTSS1 o VTSS2, y el VTSS receptor no es configurable. Este es un concepto muy importante que se debe entender, porque actualmente no hay parámetros de clase de gestión que seleccionen un VTSS específico para dirigir un VTV. Esta configuración no es útil para implementación en un entorno de ubicaciones principal y secundaria, donde el VTV debe terminar en una ubicación secundaria específica y puede transformar las configuraciones bidireccionales ampliadas en configuraciones no deseadas. La replicación sincrónica y la asincrónica están disponibles para su uso en el VTSS emisor. Cada VTSS debe tener RTS equivalentes (modelo similar) conectadas.

La configuración 2 adquiere una importancia superior cuando se decide implementar la replicación bidireccional en un entorno de agrupación en clusters ampliada. Si se requiere la replicación bidireccional, use la configuración "muchos VTSS a un VTSS" en una dirección y la configuración "VTSS par" en la otra dirección, donde el "VTSS par" se configura entre los dos VTSS en que debe residir el VTV replicado.

Como se muestra en las sentencias *CONFIG* para la configuración 2 que se muestra a continuación:

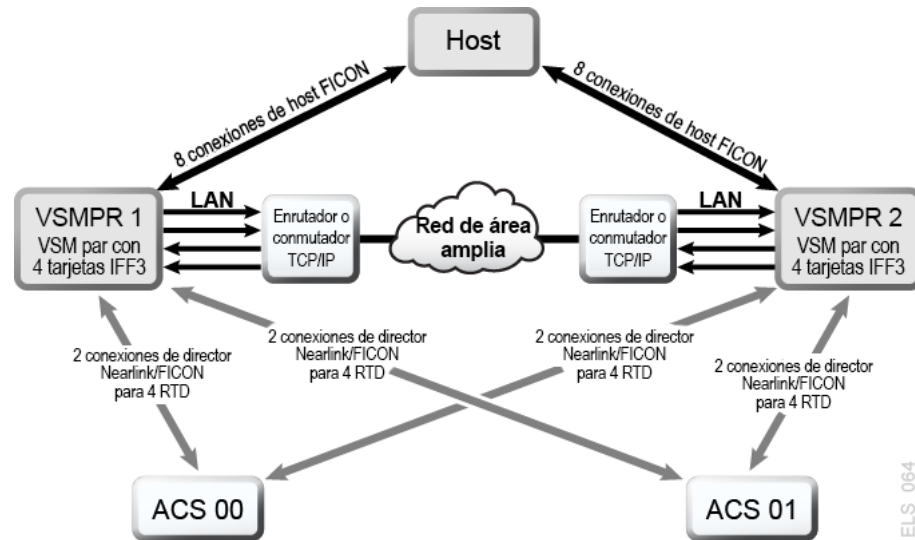
- La sentencia *CLUSTER* define todos los nombres de VTSS configurados para agrupación en clusters.
- La sentencia *CLINK* define la ubicación del puerto Nearlink en el VTSS emisor y su *PAR* o VTSS de destino.

```
//CREATCFG EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCNTL DD DSN=h1q.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=h1q.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=h1q.DBASETBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA11A00 DEVNO=1A00 CHANIF=0C
RTD NAME=PA11A01 DEVNO=1A01 CHANIF=0D
```

```
RTD NAME=PA11A02 DEVNO=1A02 CHANIF=0K
RTD NAME=PA11A03 DEVNO=1A03 CHANIF=0L
RTD NAME=PA12A08 DEVNO=2A08 CHANIF=1C
RTD NAME=PA12A09 DEVNO=2A09 CHANIF=1D
RTD NAME=PA12A0A DEVNO=2A0A CHANIF=1K
RTD NAME=PA12A0B DEVNO=2A0B CHANIF=1L
VTD LOW=7900 HIGH=79FF
VTSS NAME=VTSS2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA23A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PA23A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PA23A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PA23A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PA24A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PA24A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PA24A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PA24A0B DEVNO=4A0B CHANIF=1L
VTD LOW=8900 HIGH=89FF
VTSS NAME=VTSS3 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA33A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PA33A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PA33A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PA33A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PA34A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PA34A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PA34A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PA34A0B DEVNO=4A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSS(VTSS1,VTSS2,VTSS3)
CLINK VTSS=VTSS3 CHANIF=0G PART=VTSS1
CLINK VTSS=VTSS3 CHANIF=0G PART=VTSS2
```

Cluster VSM5 a VSM5 con CLINK TCP/IP

Figura A.7. VSM5 en cluster con CLINK TCP/IP



En [Figura A.7, “VSM5 en cluster con CLINK TCP/IP”](#), se muestra un ejemplo de un cluster VSM5 a VSM5 con CLINK TCP/IP.

En [Figura A.7, “VSM5 en cluster con CLINK TCP/IP”](#), suponga que, por motivos de redundancia, usa destinos en tarjetas IFF separadas en cada VSM5 para IP nativo, como se muestra en [Tabla A.1, “Valores de IPIF de CLINK para VSMPR1”](#) and [Tabla A.2, “Valores de IPIF de CLINK para VSMPR2”](#).

Tabla A.1. Valores de IPIF de CLINK para VSMPR1

Tarjeta IFF	Número de destino	IP de muestra	IPIF de CLINK correspondiente
IFF0	Destino 0	128.0.1.1	0A:0
IFF1	Destino 0	128.0.2.1	0I:0
IFF2	Destino 0	128.0.3.1	1A:0
IFF3	Destino 0	128.0.4.1	1I:0

Tabla A.2. Valores de IPIF de CLINK para VSMPR2

Tarjeta IFF	Número de destino	IP de muestra	IPIF de CLINK correspondiente
IFF0	Destino 0	128.0.1.2	0A:0
IFF1	Destino 0	128.0.2.2	0I:0
IFF2	Destino 0	128.0.3.2	1A:0
IFF3	Destino 0	128.0.4.2	1I:0

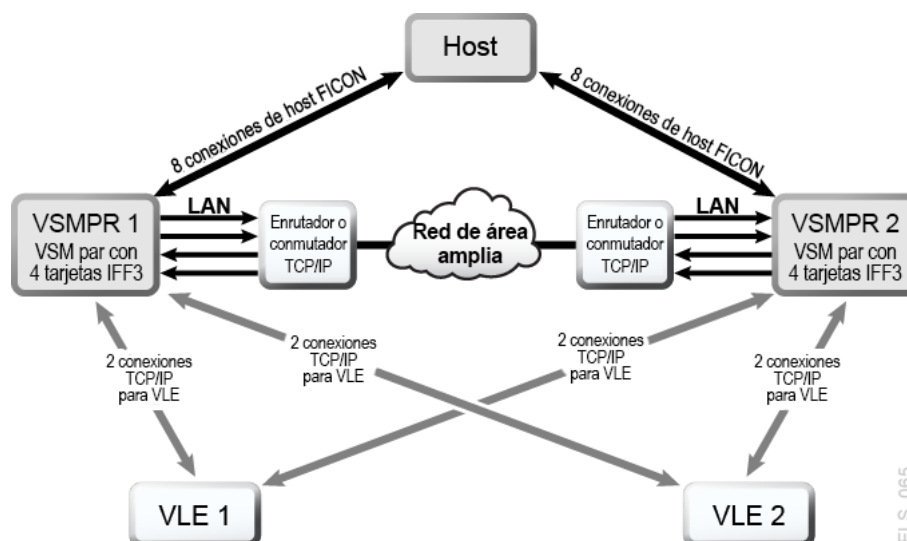
En el ejemplo que aparece a continuación, se muestra CONFIG JCL para definir la configuración que se muestra en [Figura A.7, “VSM5 en cluster con CLINK TCP/IP”](#) con los valores que se muestran en [Tabla A.1, “Valores de IPIF de CLINK para VSMPR1”](#) y [Tabla A.2, “Valores de IPIF de CLINK para VSMPR2”](#).

```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCNTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASETBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MVCFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTVPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=8900 HIGH=89FF
RTD NAME=VPR12A00 DEVNO=2A00 CHANIF=0C:0
RTD NAME=VPR12A01 DEVNO=2A01 CHANIF=0C:1
RTD NAME=VPR12A02 DEVNO=2A02 CHANIF=0C:2
RTD NAME=VPR12A03 DEVNO=2A03 CHANIF=0C:3
RTD NAME=VPR12A04 DEVNO=2A04 CHANIF=0G:0
RTD NAME=VPR12A05 DEVNO=2A05 CHANIF=0G:1
RTD NAME=VPR12A06 DEVNO=2A06 CHANIF=0G:2
RTD NAME=VPR12A07 DEVNO=2A07 CHANIF=0G:3
VTSS NAME=VSMPR2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=9900 HIGH=99FF
RTD NAME=VPR22B00 DEVNO=2B00 CHANIF=0C:0
RTD NAME=VPR22B01 DEVNO=2B01 CHANIF=0C:1
RTD NAME=VPR22B02 DEVNO=2B02 CHANIF=0C:2
RTD NAME=VPR22B03 DEVNO=2B03 CHANIF=0C:3
RTD NAME=VPR22B04 DEVNO=2B04 CHANIF=0G:0
RTD NAME=VPR22B05 DEVNO=2B05 CHANIF=0G:1
RTD NAME=VPR22B06 DEVNO=2B06 CHANIF=0G:2
RTD NAME=VPR22B07 DEVNO=2B07 CHANIF=0G:3
CLUSTER NAME=CLUSTER1 VTSSs(VSMPR1, VSMPR2)
CLINK VTSS=VSMPR1 IPIF=0A:0
CLINK VTSS=VSMPR1 IPIF=0I:0
CLINK VTSS=VSMPR1 IPIF=1A:0
CLINK VTSS=VSMPR1 IPIF=1I:0
CLINK VTSS=VSMPR2 IPIF=0A:0
CLINK VTSS=VSMPR2 IPIF=0I:0
CLINK VTSS=VSMPR2 IPIF=1A:0
CLINK VTSS=VSMPR2 IPIF=1I:0
```

Cluster VSM5 a VSM 6 con CLINK TCP/IP y VLE interconectadas

En [Figura A.8, “ejemplo de cluster VSM5 a VSM6 con CLINK TCP/IP y VLE interconectadas”](#), se muestra un ejemplo de un cluster VSM5 a VSM 6 con CLINK TCP/IP, donde cada VTSS están interconectado con dos VLE.

Figura A.8. ejemplo de cluster VSM5 a VSM6 con CLINK TCP/IP y VLE interconectadas



En Figura A.8, “ejemplo de cluster VSM5 a VSM6 con CLINK TCP/IP y VLE interconectadas”, suponga que, por motivos de redundancia, usa destinos en tarjetas IFF separadas para el VSM5 (VSMPR1) para IP nativo y para las conexiones de la VLE, como se muestra en Tabla A.3, “Valores de IPIF de CLINK para VSMPR1” and Tabla A.4, “Valores de IPIF de RTD para VSMPR1”.

Tabla A.3. Valores de IPIF de CLINK para VSMPR1

Tarjeta IFF	Número de destino	IP de muestra	IPIF de CLINK correspondiente
IFF0	Destino 0	128.0.1.1	0A:0
IFF1	Destino 0	128.0.2.1	0I:0
IFF2	Destino 0	128.0.3.1	1A:0
IFF3	Destino 0	128.0.4.1	1I:0

Tabla A.4. Valores de IPIF de RTD para VSMPR1

Tarjeta IFF	Número de destino	IP de muestra	IPIF de CLINK correspondiente
IFF0	Destino 1	128.0.1.2	0A:1
IFF1	Destino 1	128.0.2.2	0I:1
IFF2	Destino 1	128.0.3.2	1A:1
IFF3	Destino 1	128.0.4.2	1I:1

En el ejemplo que aparece a continuación, se muestra CONFIG JCL para definir la configuración que se muestra en Figura A.8, “ejemplo de cluster VSM5 a VSM6 con CLINK TCP/IP y VLE interconectadas” con los valores que se muestran en Tabla A.3, “Valores de IPIF de CLINK para VSMPR1” y Tabla A.4, “Valores de IPIF de RTD para VSMPR1”.

```

//CREATECF EXEC PGM=SLUADMIN, PARM=' MIXED '
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASEBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MVCFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTVPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=9900 HIGH=99FF
RTD NAME=VL1RTD1 STORMNGR=VLE1 IPIF=0A:1
RTD NAME=VL1RTD2 STORMNGR=VLE1 IPIF=0I:1
RTD NAME=VL2RTD1 STORMNGR=VLE2 IPIF=1A:1
RTD NAME=VL2RTD2 STORMNGR=VLE2 IPIF=1I:1
CLUSTER NAME=CLUSTER1 VTSSs(VSMPR1, VSMPR2)
CLINK VTSS=VSMPR1 IPIF=0A:0
CLINK VTSS=VSMPR1 IPIF=0I:0
CLINK VTSS=VSMPR1 IPIF=1A:0
CLINK VTSS=VSMPR1 IPIF=1I:0
CLINK VTSS=VSMPR2 IPIF=0A:0
CLINK VTSS=VSMPR2 IPIF=0I:0
CLINK VTSS=VSMPR2 IPIF=1A:0
CLINK VTSS=VSMPR2 IPIF=1I:0

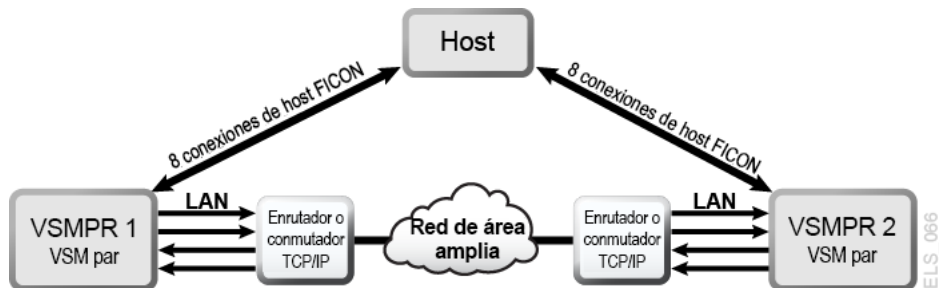
```

En este ejemplo, tenga en cuenta que, si bien los valores de los parámetros *CLINK IPIF* y *RTD IPIF* para el VSM5 (VSMPR1) deben coincidir con los valores que se muestran en [Tabla A.3, “Valores de IPIF de CLINK para VSMPR1”](#) y [Tabla A.4, “Valores de IPIF de RTD para VSMPR1”](#), los valores de *CLINK IPIF* y *RTD IPIF* para el VSM 6 (VSMPR2) solo deben cumplir con las restricciones de VTCS sobre estos valores y deben ser únicos para cada VTSS; no corresponden a un valor real de los puertos TCP/IP de VSM 6.

Cluster VSM 6 a VSM 6 “sin cinta” con CLINK TCP/IP

En [Figura A.9, “ejemplo de Cluster VSM6 a VSM6 sin cinta con CLINK TCP/IP”](#), se muestra un ejemplo de Cluster VSM 6 a VSM 6 sin cinta con CLINK TCP/IP.

Figura A.9. ejemplo de Cluster VSM6 a VSM6 sin cinta con CLINK TCP/IP



En el ejemplo que aparece a continuación, se muestra CONFIG JCL para definir la configuración que se muestra en [Figura A.9, “ejemplo de Cluster VSM6 a VSM6 sin cinta con CLINK TCP/IP”](#).

```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASEBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MCVFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTVPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=8900 HIGH=89FF
CLUSTER NAME=CLUSTER1 VTSSs(VSMR1, VSMR2)
CLINK VTSS=VSMR1 IPIF=0A:0
CLINK VTSS=VSMR1 IPIF=0A:1
CLINK VTSS=VSMR1 IPIF=0A:2
CLINK VTSS=VSMR1 IPIF=0A:3
CLINK VTSS=VSMR2 IPIF=0A:0
CLINK VTSS=VSMR2 IPIF=0A:1
CLINK VTSS=VSMR2 IPIF=0A:2
CLINK VTSS=VSMR2 IPIF=0A:3
```

En este ejemplo, tenga en cuenta que los valores de *CLINK IPIF* para ambos VSM6 solo deben cumplir con las restricciones de VTCS sobre estos valores y deben ser únicos para cada VTSS; no corresponden a un valor real de los puertos TCP/IP de VSM 6. También tenga en cuenta que, dado que el cluster es sin cinta, no hay sentencias *CONFIG RTD* para ninguno de los VSM 6.

¿Debe usar una configuración unidireccional o bidireccional?

Puede usar las sentencias *VTSSLST* y *VTSSSEL* para transformar un cluster bidireccional en uno unidireccional. ¿Por qué querría hacer esto? ¿Qué sucedería si deseara cambiar los roles de los VTSS principal y secundario? Comience realizando la misma configuración que se describe en el procedimiento que se indica en "[Configuración y gestión de un sistema de VTSS en cluster bidireccional](#)." Después de finalizar el paso 5, debe realizar un cambio con las siguientes sentencias *VTSSLST* y *VTSSSEL*.

```
VTSSLST NAME(SITEA) VTSS(VSMR1)
VTSSSEL FUNCTION(SCRATCH) HOST(MVSA) VTSSLST(SITEA)
VTSSSEL FUNCTION(SPECIFIC) HOST(MVSA) VTSSLST(SITEA)
```

En este ejemplo:

- La sentencia *VTSSLST* define el SITEA de la lista de VTSS que contiene solo VSMR1.
- Las sentencias *VTSSSEL* dirigen montajes de VTV específicos y reutilizables de MVSA a SITEA, que contiene **solo** VSMR1, lo que lo convierte en el principal de manera eficaz.

Por lo tanto, este cluster es, en realidad, bidireccional, pero las sentencias *VTSSLST* y *VTSSSEL* nos dan la flexibilidad de definir cualquier VTSS como el principal y el otro como secundario, simplemente, cargando las correspondientes sentencias de control *MGMTclas*, *STORclas*, *VTSSLST* y *VTSSSEL* con el comando *MGMTDEF*.

¿Qué sucede si desea cambiar el principal y el secundario? En este caso, reescriba las sentencias *VTSSLST* y *VTSSSEL* para que VSMR2 pase a ser el principal y VSMR1, el secundario.

```
VTSSLST NAME(SITEB) VTSS(VSMR2)
VTSSSEL FUNCTION(SCRATCH) HOST(MVSB) VTSSLST(SITEB)
VTSSSEL FUNCTION(SPECIFIC) HOST(MVSB) VTSSLST(SITEB)
```

En este ejemplo:

- La sentencia *VTSSLST* define el SITEB de la lista de VTSS que contiene solo VSMR2.
- Las sentencias *VTSSSEL* dirigen montajes de VTV específicos y reutilizables de MVSB a SITEB, que contiene **solo** VSMR2, lo que lo convierte, efectivamente, en el principal.

Finalmente, ¿qué sucedería si llegara el momento en que todo funcionara mejor con este cluster como cluster bidireccional? En este caso, suprima las sentencias *VTSSLST* y *VTSSSEL* y vuelva a cargar las definiciones.

Glosario

Nota:

Las entradas del glosario que están designadas como (I) corresponden al diccionario *IBM Dictionary of Computing*.

4410	Módulo de almacenamiento de biblioteca (LSM) estándar StorageTek de Oracle.
4480	Transporte de cartuchos de media pulgada y 18 pistas StorageTek de Oracle.
4490	Transporte de cartuchos de cinta larga de 36 pistas StorageTek de Oracle compatibles con ESCON. Se conoce también como Silverton.
9310	Módulo de almacenamiento de biblioteca (LSM) StorageTek de Oracle, versión de alto rendimiento de 4410 LSM. Se conoce también como PowderHorn.
9360	Módulo de almacenamiento de biblioteca (LSM) StorageTek de Oracle. Se conoce también como WolfCreek.
9740	Módulo de almacenamiento de biblioteca (LSM) StorageTek de Oracle. Se conoce también como TimberWolf.
ACS	Sistema de cartuchos automatizado (ACS, Automated Cartridge System). Un subsistema completamente automatizado de almacenamiento de cartuchos y recuperación de bibliotecas compuesto por uno o varios módulos de almacenamiento de biblioteca (LSM) conectados mediante puertos passthru.
ACSid	Método utilizado en el proceso LIBGEN para identificar ACS mediante el uso de dígitos decimales, de 00 a 99.
ACSLs	Automated Cartridge System Library Software. Software de control de bibliotecas StorageTek de Oracle que se ejecuta en el sistema de control de bibliotecas basado en UNIX®.
adaptador de ruta de control (CPA)	Un dispositivo de hardware de Bus-Tech, Inc. que posibilita las comunicaciones entre un canal multiplexor de bloques del procesador de host y una red de área local.
adaptador de ruta de datos	Un dispositivo de hardware que realiza conversiones del protocolo de datos de un sistema informático de cliente al protocolo de datos de la unidad de control StorageTek o IMU. Un ejemplo es la interconexión TC44-AA/BA STI a 4400 ACS de DEC.
agrupación de cintas nuevas	Un subconjunto definido de todas las cintas nuevas. Las subagrupaciones están compuestas por uno o varios rangos de VOLSER con características

	similares (tipo de volumen, carrete o cartucho, tamaño del carrete, longitud, ubicación física, etc.). Algunas instalaciones también pueden subdividir sus agrupaciones nuevas según otras características, como el tipo de etiqueta.
almacenamiento virtual	Una función del sistema operativo que permite asignar los requisitos de almacenamiento principal por segmentos (o páginas) según las necesidades de los programas, lo cual crea la aparente existencia de almacenamiento ilimitado o virtual.
archivo	Un conjunto de registros relacionados tratados como una unidad.
archivo de rastreo	Un archivo que contiene información útil para depurar el sistema.
arquitectura de redes de sistemas (SNA)	Una descripción de la estructura lógica, los formatos, los protocolos y las secuencias operativas para transmitir unidades de información y controlar la configuración y el funcionamiento de las redes.
asignación	La asignación de recursos a una tarea específica.
asignación dirigida	Consulte establecimiento de prioridades de unidades.
Automated Cartridge System Library Software (ACSL)	Consulte ACSL .
base de datos de biblioteca	Un archivo o conjunto de datos que contiene información sobre la ubicación y el estado de los volúmenes de medios extraíbles, como la ubicación de las celdas y el estado de nuevo. También se denomina conjunto de datos de control (CDS).
biblioteca	Consulte TapePlex.
biblioteca ACS	Una biblioteca está compuesta por uno o varios sistemas de cartuchos automatizados (ACS), unidades de cartuchos conectadas y cartuchos que residen en los ACS.
BISYNC	Comunicaciones binarias sincrónicas (BISYNC, Binary Synchronous Communications). Un protocolo de nivel inferior desarrollado por IBM y utilizado para transmitir datos en un enlace de comunicaciones sincrónicas. Se trata de una forma de transmisión de datos en la cual la sincronización de caracteres es controlada por señales de tiempo generadas en las estaciones de envío y recepción.
bloque de control de eventos (ECB)	Proporciona un área para almacenar un código de finalización una vez que se completa una operación.
cable coaxial	Un medio de transmisión utilizado en las transmisiones de datos para redes que utilizan comunicaciones sincrónicas, en comparación con el

	par trenzado, que es el medio principal para las comunicaciones RS-232 asincrónicas.
canal	Un dispositivo que conecta el host y el almacenamiento principal con las unidades de control de dispositivos de entrada y salida. Un canal de dúplex completo tiene dos rutas (es decir, 2 cables, o un cable con señales en dos frecuencias). Un canal de dúplex parcial requiere que un puerto reciba y que el otro transmita.
canal a canal (CTC)	Se refiere a la comunicación (transferencia de datos) entre programas en lados opuestos de un adaptador de canal a canal.(I)
canal de utilidad de acoplamiento	Un canal de fibra óptica de alto ancho de banda que proporciona la conectividad de alta velocidad necesaria para el uso compartido de datos entre una utilidad de acoplamiento y los complejos del procesador central directamente conectados a ella.(I)
CAPid	Un CAPid identifica de forma exclusiva la ubicación de un CAP según el LSM donde reside. El CAPid tiene el formato "AA:LL:CC", donde AA:LL es el LSMid y CC es el número de CAP de dos dígitos.
carga inicial de programas (IPL)	Un proceso que activa el restablecimiento de un equipo.
cartucho	La carcasa plástica alrededor de una cinta. Tiene aproximadamente 100 mm (4 in) x 125 mm (5 in) x 25 mm (1 in). La cinta pasa automáticamente cuando se carga en un transporte. Hay un bloque de plástico conectado a la cinta para la alimentación automática. El lomo del cartucho contiene una etiqueta de código de barras/OCR que indica el VOLSER (identificador de volumen de cinta).
cartucho de capacidad estándar	Un cartucho que puede utilizarse en cualquier transporte longitudinal (es decir, 4480, 4490, 9490 o 9490EE).
Cartucho de capacidad mejorada	Un cartucho que tiene una longitud de 1100 ft y únicamente puede ser utilizado en transportes de 36 pistas (es decir, 4490, 9490 y 9490EE).
cartucho helicoidal	Cartucho de exploración helicoidal de alta capacidad que puede almacenar hasta 50 GB de datos no comprimidos. Este cartucho únicamente puede utilizarse en transportes RedWood (SD-3).
celda	Un receptáculo en el LSM en el que se almacena un solo cartucho.
CGI	Interfaz de puerta de enlace común.
cinta de actualización de programa (PUT)	Una o varias cintas que contienen actualizaciones, o versiones nuevas, del software del sistema MVS/CSC.

cinta nueva	Una cinta que está disponible para cualquier usuario porque no tiene propietario.
cliente	El usuario final de los servicios ACS provistos por el sistema de control de bibliotecas.
cliente/servidor	Modelo de interacción en un sistema distribuido en el cual un programa ubicado en un sitio envía una solicitud a un programa ubicado en otro sitio y espera una respuesta. El programa solicitante se denomina cliente; el programa que responde a la solicitud se denomina servidor.
código de barras	Un código compuesto por una serie de barras de distinto ancho. Este código aparece en la etiqueta externa adherida al lomo de un cartucho y es equivalente al número de serie de volumen (VOLSER). Este código es leído por el sistema de visión del robot.
códigos de error (EC)	Códigos numéricos mostrados por mensajes que indican el tipo de problema que causó un error.
complejo	Un sistema compuesto por otros sistemas, específicamente el sistema de servidor ACS y el sistema de cliente.
complejo de bibliotecas	Un complejo de bibliotecas está compuesto por un conjunto de datos de control (CDS) del HSC y puede contener hasta 256 sistemas de cartuchos automáticos (ACS), cada uno de los cuales puede contener hasta 24 módulos de almacenamiento de biblioteca (LSM).
componente de control de bibliotecas	Software que controla el montaje y el desmontaje de cartuchos en un ACS.
componente de gestión de almacenamiento (SMC)	Interfaz de software entre el sistema operativo z/OS de IBM y los sistemas de control de bibliotecas automatizadas StorageTek, HSC y MVS/CSC. El SMC se ocupa del procesamiento de asignaciones, el manejo de mensajes y el procesamiento de SMS para la solución ELS.
componente de sistema de cliente (CSC)	Software que proporciona una interfaz entre el sistema operativo del sistema informático de cliente y el sistema de control de biblioteca (LCS) StorageTek.
configuración mixta	Una configuración que contiene diferentes tipos de unidades de cartuchos en los modos manual y de biblioteca.
conjunto de datos	Un conjunto de registros tratados como una unidad.
conjunto de datos de control (CDS)	Conjunto de datos utilizados por el software de host para controlar las funciones de la biblioteca automatizada. También se denomina base de datos de biblioteca.

conmutación de servidor dinámico	La capacidad de conmutador procesadores del servidor cuando se produce un error del sistema en el servidor activo.
consola	El dispositivo de E/S principal para controlar una sesión en un sistema.
consola de operador	En este documento, la consola de operador se refiere a la consola del sistema de cliente de MVS.
consola del procesador LCS	La consola del procesador del sistema de control de bibliotecas se utiliza para controlar el sistema operativo VM (para el LCS basado en VM).
consola del sistema MVS	El MVS/CSC proporciona una interfaz de operador mediante la consola del sistema MVS.
controlador de E/S de cartucho de cinta	Software del sistema operativo que ejecuta comandos (por ejemplo, de lectura, escritura y retroceso) para los subsistemas de cartuchos. Es el punto central del software para conectar un tipo determinado de unidad de control. (Un ejemplo es el producto CARTLIB de StorageTek de Oracle).
CTC	De canal a canal (CTC, Channel-to-channel).
DFSMS	Subsistema de gestión de almacenamiento de la utilidad de datos (DFSMS, Data Facility Storage Management Subsystem).
dirección	Representación codificada del ID de hardware o el destino u origen de los datos.
dirección de Internet	El sistema de numeración utilizado para especificar una red o host en esa red para comunicaciones TCP/IP. La notación de dirección de Internet estándar tiene un formato decimal con punto.
dispositivo de almacenamiento de acceso directo (DASD)	Término de IBM para referirse a un dispositivo de almacenamiento de unidades de disco.
ECART	Cartucho de capacidad mejorada (ECART, Enhanced Capacity Cartridge).
en espera	El estado de una estación que varía en línea, pero que está conectada a la LMU en espera de un ACS de LMU dual.
enlace de cliente	El enlace de comunicaciones entre el LCS y un cliente.
Enterprise Systems Connection (ESCON)	Un conjunto de productos y servicios que proporciona un entorno conectado de forma dinámica que utiliza cables como medio de transmisión.(I)
ESCON	Enterprise Systems Connection.
establecimiento de comunicación	Una señal de flujo de control enviada de un proceso a otro.

establecimiento de prioridades de unidades	(Anteriormente conocida como asignación dirigida). Se refiere a la función del SMC de influir en la selección de una unidad determinada sobre la base de criterios de asignación, como la ubicación del volumen.
estación	Una ruta de hardware entre el equipo host y una LMU, a través de la cual VM/HSC y la LMU envían información de control.
Ethernet	Una arquitectura LAN que utiliza una topología de bus que permite que una variedad de equipos se conecten a un cable coaxial blindado común. La arquitectura Ethernet es similar al estándar IEEE 802.3.
etiqueta OCR	Etiqueta de reconocimiento óptico de caracteres. Una etiqueta externa adherida al lomo de un cartucho, que es legible en lenguaje natural y por una máquina.
exclusión de unidades	(Anteriormente conocida como separación de dispositivos). Se refiere a la función de excluir unidades para una solicitud de asignación sobre la base de criterios de exclusión del SMC.
finalización anormal de tarea	Un problema de software o hardware que finaliza una tarea de procesamiento informática.
GB	1073.741.834 bytes de almacenamiento.
gestor de almacenamiento virtual (VSM)	Una solución de almacenamiento que virtualiza volúmenes y transportes en un buffer de VTSS para mejorar el uso de medios y transportes.
host	Computadora que controla una red de computadoras.
HTTP	Protocolo de transferencia de hipertexto.
ICRC	Capacidad mejorada de registro de cartucho. Una función de compresión y compactación que aumenta la cantidad de datos que pueden almacenarse en un cartucho de media pulgada.
ID de usuario	El ID de usuario, a veces conocido como ID de usuario de VM, es el nombre que identifica el cliente o usuario de la "máquina virtual" específica.
IEEE 802.3	Un estándar creado por el IEEE y aceptado en todo el mundo para las redes de área local que utilizan CSMA/CD (acceso múltiple por detección de portadora con detección de colisiones).
Internet	Una recopilación de redes que utiliza TCP/IP y que funciona como una red virtual.
ISMF	Utilidad de gestión de almacenamiento interactiva.
JES	Subsistema de entrada de trabajos.(I)

JES2	Un subsistema de MVS que recibe los trabajos en el sistema, los convierte al formato interno, los selecciona para su ejecución, procesa el resultado y los depura del sistema. En una instalación con más de un procesador, cada procesador JES2 controla de forma independiente su procesamiento de entrada, programación y salida del trabajo. Consulte también JES3.(I)
JES3	Un subsistema de MVS que recibe los trabajos en el sistema, los convierte al formato interno, los selecciona para su ejecución, procesa el resultado y los depura del sistema. En complejos que tienen varias unidades de procesamiento acopladas de forma deficiente, el programa JES3 administra los procesadores de modo que el procesador global tenga un control centralizado de los procesadores locales y distribuya los trabajos mediante una cola de trabajo común. Consulte también JES2.(I)
LAN	Red de área local (LAN, Local Area Network). Una red en una pequeña área geográfica (local).
LAN sincrónica	Red de área local construida en comunicaciones sincrónicas.
LCS	Sistema de control de bibliotecas (LCS, Library Control System).
LCU	Consulte unidad de control de bibliotecas.
lenguaje de control de trabajos (JCL)	Lenguaje orientado a la resolución de problemas diseñado para describir los requerimientos de procesamiento de un trabajo para un sistema operativo.
LIBGEN	El proceso de definir la configuración de una biblioteca para VM/HSC.
LMU	Unidad de gestión de bibliotecas (LMU, Library Management Unit). Un producto de hardware y software que coordina las actividades de uno o varios LSM/LCU.
LMU dual	Una función de hardware/microcódigo que proporciona una capacidad de LMU redundante.
LMU en espera	La LMU redundante en una configuración de LMU dual que está lista para tomar el control en caso de error de la LMU principal o cuando el operador ejecuta un comando SWITCH.
LMU principal	El LMU que controla actualmente el trabajo funcional del ACS en una configuración de LMU dual.
LP	Puerto lógico. Software de CLS que interactúa con el sistema de cliente. El CLSLP es uno de los componentes de software utilizados para transferir datos entre el sistema de cliente y VM/HSC.
LSM	Módulo de almacenamiento de biblioteca (LSM, Library Storage Module). El LSM estándar (4410); una estructura de 12 lados con espacio de

	almacenamiento para hasta 6000 cartuchos. También contiene un robot independiente con visión asistida que mueve los cartuchos entre las celdas de almacenamiento y los transportes conectados. Consulte también PowderHorn, SL3000, SL8500 y WolfCreek.
LSMid	Un LSMid está compuesto por el ACSid unido (o concatenado) al número de LSM.
máquina virtual (VM)	Consulte VM .
método de acceso virtual de telecomunicaciones (VTAM)	Software de comunicaciones residente en el host, desarrollado por IBM, que sirve como interfaz común para las comunicaciones.
módem	Un dispositivo que permite la transmisión de datos digitales a través de una utilidad de transmisión análoga.
modo automático	Una relación entre un LSM y todos los hosts conectados. Los LSM que funcionan en modo automático manejan los cartuchos sin intervención del operador. Este es el modo de funcionamiento normal de un LSM que fue modificado en línea. La situación opuesta es el "modo manual". Consulte modo manual.
modo de biblioteca	El funcionamiento de un sistema de cartuchos 4480 como parte de un sistema de cartuchos automatizado 4400, a diferencia del modo manual, en el cual el operador inserta cartuchos en los transportes. Consulte modo manual.
modo de conexión	Una relación entre un host y un ACS. En este modo, el host y un ACS son capaces de comunicarse (es decir, hay al menos una estación en línea a este ACS).
modo de desconexión	Una relación entre un host y un ACS. En este modo, el host y el ACS no son capaces de comunicarse (no hay estaciones en línea a este ACS).
modo manual	Funcionamiento de una unidad de cartuchos independientemente de un ACS. Consulte modo de biblioteca.
módulo de almacenamiento de biblioteca (LSM)	Consulte LSM .
multicliente	El entorno donde más de un sistema de cliente (homogéneo o heterogéneo) se conecta a un LCS.
nombre esotérico	El nombre asignado a transportes que tienen el mismo tipo de dispositivo.
notación decimal con punto	La representación sintáctica de un valor entero de 32 bits que consta de cuatro números de 8 bits escritos en base 10 con puntos que los separan. En

	las descripciones de TCP/IP, la notación decimal con punto se utiliza para direcciones de Internet.
número de conexión	El identificador único del servidor para una ruta de comunicaciones. El TCP/IP asigna el número para identificar la conexión única entre el nodo de servidor y un puerto específico en el servidor, y el nodo de cliente y un puerto específico en el cliente. El número de conexión existe únicamente si existe la conexión.
número de dispositivo	Un número hexadecimal de cuatro dígitos que identifica de forma exclusiva a un dispositivo conectado a un procesador.
número de LSM	Un método utilizado para identificar un LSM. Un número de LSM es el resultado de la definición del parámetro LSM de la macro SLIACS durante LIBGEN. El primer LSM enumerado en este parámetro adquiere el número 00 (decimal), el segundo LSM enumerado adquiere el número 01 y así sucesivamente hasta que se hayan identificado todos los LSM (hasta un máximo de 99).
número de serie de volumen (VOLSER)	Un identificador de un volumen físico.
panel de unidades	Una pared de LSM que contiene transportes de cinta. El panel de unidades para un transporte T9840 puede contener 10 o 20 transportes. El panel de unidades para un transporte que no sea T9840 puede contener un máximo de cuatro transportes.
paquete preconfigurado	Un paquete de servidor de almacenamiento que incluye todos los valores de parámetros de configuración, software y hardware suministrados por el proveedor.
plataforma de control de bibliotecas	Hardware y software que proporcionan el entorno adecuado para el sistema de control de bibliotecas.
PowderHorn (9310)	La versión de alto rendimiento del LSM estándar.
preferencia de dispositivos	El proceso de preferir un tipo de transporte de 36 pistas en lugar de otro tipo de transporte de 36 pistas.
procedimientos de recuperación de errores (ERP)	Procedimientos diseñados para ayudar a aislar errores del equipo y, si es posible, recuperarse de ellos.
procesador de control de bibliotecas	Hardware informático adecuadamente configurado que permite el funcionamiento del sistema de control de bibliotecas.
protocolo	Una descripción formal de formatos de mensajes y las reglas que deben seguir dos o varios equipos para intercambiar esos mensajes.

protocolo de control de transmisión (TCP)	Un protocolo estándar entre redes que proporciona un servicio de transmisión de dúplex completo.
protocolo de Internet (IP)	Descripción formal de mensajes y reglas que dos redes utilizan para intercambiar mensajes.
protocolo de transferencia de archivos (FTP)	Un comando TCP/IP que ofrece una manera de transferir archivos entre equipos conectados a través de TCP/IP.
puerto de acceso de cartuchos (CAP)	Una unidad que permite la inserción o la expulsión de varios cartuchos en un LSM sin intervención humana en el LSM.
puerto físico	Hardware de comunicaciones necesario para admitir un enlace de servidor/cliente.
puerto local	La designación de una aplicación o un proceso determinados entre varios disponibles para un procesador de host compatible con TCP/IP.
puerto lógico (LP)	Consulte LP .
puerto passthru (PTP)	Un mecanismo que permite que un cartucho se transfiera de un LSM a otro en un ACS de varios LSM.
recuperación	Procedimientos manuales o automáticos para resolver problemas en el sistema de servidor.
red de área local (LAN)	Consulte LAN .
reel-id	Identificador de un volumen de cinta específico. Es equivalente al número de serie de volumen (VOLSER).
SD-3	Transporte de cartuchos helicoidales StorageTek de Oracle. También conocido como RedWood.
separación de dispositivos	Consulte exclusión de unidades .
servidor	Un sistema de control de bibliotecas ELS, como HSC. En un SMC, un servidor está representado por una ruta SERVER designada a un TAPEPLEX designado. A pesar de que el componente de servidor HTTP del SMC es necesario como el middleware en el host remoto, el servidor, en lo que se refiere al SMC, es el sistema de control de bibliotecas ELS que funciona en el host remoto.
servidor de almacenamiento	Un conjunto de productos de hardware y software diseñado para permitir que sistemas informáticos heterogéneos usen servicios de bibliotecas de cartuchos de cintas automatizadas.
sincrónico	Consulte BISYNC .

sistema de cartuchos automatizado (ACS)	Consulte ACS .
sistema de cliente	El sistema para el cual el LCS proporciona una interfaz para StorageTek Automated Cartridge System.
sistema de control de bibliotecas (LCS)	La plataforma de control de bibliotecas y el software de control de bibliotecas.
sistema de control de cinta virtual (VTCS)	El código de host principal para la solución del gestor de almacenamiento virtual (VSM). Este código funciona en un espacio de direcciones separado, pero se comunica estrechamente con el HSC.
sistema de gestión de bibliotecas de cintas (TLMS)	Un TLMS, de la manera en que se utiliza en este documento, se refiere a cualquier sistema de gestión de bibliotecas de cintas, no a CA-1.
sistema operativo (SO)	Software que controla la ejecución de programas que facilitan el funcionamiento general del sistema.
SL3000	La biblioteca modular StorageTek SL3000 de Oracle ofrece funciones de partición física y lógica, de medios combinados, gestión avanzada y alta disponibilidad. Admite entornos combinados, incluidos sistemas abiertos y mainframe, y el número de ranuras de cartuchos se puede incrementar de 200 a 6000.
SL8500	La biblioteca modular StorageTek SL8500 de Oracle ofrece funciones de partición física y lógica, de medios combinados, gestión avanzada y alta disponibilidad. Admite entornos combinados, incluidos sistemas abiertos y mainframe, y el número de ranuras de cartuchos en una configuración compleja se puede incrementar de 1450 ranuras de cartuchos estándar a 100 .800 ranuras.
socket	Una dirección única en una red, además de una dirección de nodo y el ID de una aplicación específica en una red específica. Una abstracción utilizada por TCP/IP.
socket externo	Uno de dos puntos finales en un protocolo orientado a conexión TCP/IP. Especifica la dirección de un host externo que puede conectarse al servidor.
socket local	La combinación de direcciones de una dirección de red de host compatible con TCP/IP y un puerto específico para un proceso de aplicación.
software de control de bibliotecas	Un componente de control de bibliotecas, la interfaz del sistema de cliente y utilidades de bibliotecas.
solicitud	Término utilizado para referirse a los comandos ejecutados en ACS 4400 para realizar una función relacionada con cintas.

solicitud de cambio de productos (PCR)	Una solicitud para la mejora de un producto. Generalmente, esta solicitud es realizada por un cliente, pero puede ser realizada por Oracle.
solución temporal de programa (PTF)	Una versión de software diseñada para corregir uno o varios defectos.
subsistema de almacenamiento de cinta virtual (VTSS)	El buffer del DASD que contiene volúmenes virtuales (VTV) y unidades virtuales (VTD). El VTSS es un dispositivo de hardware RAID 6 StorageTek con un microcódigo que permite la emulación de transporte. El dispositivo RAID puede leer y escribir datos de cinta desde o hacia un disco, y puede leer y escribir datos desde y hacia una unidad de cinta real (RTD).
switchover	La suposición de la función de la LMU principal por parte de la LMU en espera.
T10000A	Transporte para cartuchos de alta capacidad T10000 A StorageTek de Oracle, capaz de leer y escribir cartuchos T10000A de 120 GB o 500 GB.
T10000B	Transporte para cartuchos de alta capacidad T10000 B StorageTek de Oracle, capaz de leer y escribir cartuchos T10000B de 240 GB o 1 TB.
T10000C	Unidad de cinta de alta velocidad/capacidad T10000 C StorageTek de Oracle, que ofrece hasta 252 MB/s y 5 TB de capacidad nativa, lo que hace que sea ideal para operaciones de centros de datos con crecientes volúmenes de datos.
T10000D	Unidad de cinta de alta velocidad/capacidad T10000D StorageTek de Oracle, que ofrece hasta 252 MB/s y 8,5 TB de capacidad nativa, lo que hace que sea ideal para operaciones de centros de datos con requisitos crecientes de retención de datos.
T9840A	Transporte para cartuchos StorageTek de Oracle, centrado en el acceso, capaz de leer y escribir cartuchos 9840A.
T9840B	Transporte para cartuchos StorageTek de Oracle, centrado en el acceso, capaz de leer y escribir cartuchos T9840B.
T9840C	Transporte para cartuchos StorageTek de Oracle, centrado en el acceso, capaz de leer y escribir cartuchos T9840C.
T9840D	Transporte para cartuchos StorageTek de Oracle, centrado en el acceso, capaz de leer y escribir cartuchos T9840D.
T9940A	Transporte para cartuchos StorageTek de Oracle, centrado en la capacidad, capaz de leer y escribir cartuchos T9940A de 60 GB.
T9940B	Transporte para cartuchos StorageTek de Oracle, centrado en la capacidad, capaz de leer y escribir cartuchos T9940B de 200 GB.

TapePlex	(Anteriormente "biblioteca"). Una única configuración de hardware de StorageTek, generalmente representada por un solo conjunto de datos de control (CDS) del HSC. Un TapePlex puede contener varios sistemas de cartuchos automatizados (ACS) y subsistemas de almacenamiento de cinta virtual (VTSS).
TCP/IP	Protocolo de control de transmisión (TCP, Transmission Control Protocol)/ protocolo de Internet (IP, Internet Protocol).
tipo de evento de rastreo	Tipos de eventos rastreados por el sistema cuando el rastreo está activado.
transacción	Un conjunto específico de entradas que activa la ejecución de un proceso específico.
transmisión asincrónica	Transmisión de datos orientada a caracteres (a diferencia de la transmisión en modo de bloques de IBM).
transporte	Un dispositivo electromecánico utilizado para la alimentación, el posicionamiento y la lectura o escritura de una cinta.
transporte de cartucho de biblioteca	Consulte transporte.
UCB	Bloque de control de unidades (UCB, Unit Control Block).
unidad de biblioteca	Una unidad de cartuchos en el ACS, distinta de una unidad de cartuchos independiente.
unidad de cartuchos (CD)	Un dispositivo de hardware que contiene dos o cuatro transportes de cartuchos y fuentes neumáticas y de alimentación asociadas.
unidad de cinta	Un dispositivo de procesamiento de cintas compuesto por hasta cuatro transportes en un armario. Una unidad puede hacer referencia a un transporte individual.
unidad de control (CU)	Una unidad basada en microprocesador ubicada localmente entre un canal y un dispositivo de E/S. Convierte comandos del canal en comandos del dispositivo y envía el estado del dispositivo al canal.
unidad de control de bibliotecas (LCS)	La parte de un LSM que controla los movimientos del robot.
unidad de gestión de bibliotecas (LMU)	Consulte LMU .
uso compartido de datos	La capacidad que tienen los programas de la aplicación o los subsistemas simultáneos para acceder directamente a los mismos datos y cambiarlos sin alterar su integridad.(I)

utilidad	Programa que realiza una función secundaria a las funciones principales de un sistema informático.
utilidad de acoplamiento	Una partición lógica especial que ofrece almacenamiento en caché de alta velocidad, procesamiento de listas y funciones de bloqueo en un sysplex.(I)
Utilidad de gestión de almacenamiento interactiva.	Una serie de aplicaciones para definir clases y grupos de almacenamiento de DFSMS/MVS.
VM	Máquina virtual (VM, Virtual Machine). Una simulación funcional de una computadora y los dispositivos asociados. Cada máquina virtual es controlada por un sistema operativo adecuado.
VM/SP o VM/XA	Un sistema operativo de IBM Corporation que consta, principalmente, de dos componentes principales: CP y CMS.
volcado	Una representación impresa del contenido del almacenamiento principal en un momento t. Esta representación se utiliza para la depuración.
volumen	Un cartucho de cinta (portador de datos) que se monta o desmonta como una unidad.
volumen físico	Una unidad de datos físicamente vinculada a un medio de archivos de datos. Consulte cartucho.
WolfCreek (9360)	El LSM de alto rendimiento, con una capacidad menor que la del LSM estándar.
ZCART	Un cartucho mejorado que utiliza medios más pequeños para ofrecer el doble de capacidad que el cartucho de capacidad mejorada (ECART). Este cartucho tiene una longitud de 2200 ft y únicamente puede utilizarse en transportes TimberLine 9490EE de 36 pistas.

Índice

A

actualización de la plataforma VTCS CONFIG para definir una biblioteca remota, 82
agrupación en clusters ampliada, 94
almacenamiento de MVC para DR, 50
almacenamiento de MVC para LTR, 60
almacenamiento externo de ELS
 preparación para, 47
alta disponibilidad de datos (D-HA)
 cinta física de alta disponibilidad, 25
 cinta virtual de alta disponibilidad, 26
 planificación para la, 24

B

biblioteca remota
 actualización de la plataforma VTCS CONFIG para definir una biblioteca remota, 82
 configuración, 81
 consideraciones sobre la agrupación del MVS, 83
 modificación del archivo SMC SCMDS, 81

C

CDRT
 consideraciones sobre metadatos, 100
 ejecución de una prueba de DR, 110
 escenarios operativos, 113
 gestión de recursos, 102
 limpieza después de una prueba de DR, 112
 optimización del acceso a recursos de prueba y producción, 109
 origen de datos del VTV, 101
 políticas de VTCS, 108
 uso del software, 99
cluster bidireccional
 ejemplo de configuración y gestión, 149
 ejemplo de transformación en cluster unidireccional, 161
cluster unidireccional
 ejemplo de configuración y gestión, 143
 ejemplo de VTSS, 141
clusters bidireccionales, 92
 ejemplo de VTSS, 146

clusters unidireccionales, 91
configuración de una biblioteca remota, 81
configuraciones de VTSS en cluster
 agrupación en clusters ampliada, 94
 agrupación en clusters con conexiones TCP/IP, 97
 bidireccionales, 92
 cómo funcionan, 88
 definición, 85
 replicación sincrónica o asincrónica, 95
 requisitos, 86
 unidireccionales, 91
 uso, 85
consideraciones sobre la agrupación de MVC para bibliotecas remotas, 83

D

definición del objetivo de punto de recuperación (RTO), 21
definición del objetivo de tiempo de recuperación (RTO), 20

E

ejemplo de agrupación en clusters ampliada, 153
ejemplo de cluster VSM5 a VSM5 cluster con CLINK TCP/IP, 157
ejemplo de cluster VSM5 a VSM6 con CLINK TCP/IP y VLE interconectadas, 158
ejemplo de Cluster VSM6 a VSM6 sin cinta con CLINK TCP/IP, 160
exportación e importación de MVC por clase de almacenamiento, 41
exportación e importación de MVC por clase de gestión, 38
expulsión de volúmenes específicos a un almacén local (planta), 61

G

gestión de recursos de CDRT, 102

I

implementación de la replicación sincrónica, 95
implementación de replicación sincrónica con supervisión de trabajos, 96
importación y exportación de MVC por clase de almacenamiento, 41

importación y exportación de MVC por clase de gestión, 38

M

manejo de interrupciones temporales, 22
modificación del archivo SMC SCMDS, 81

P

planificación de pruebas de DR, 30
planificación para la alta disponibilidad de datos (D-HA), 24
políticas de VTCS para CDRT, 108
preparación para almacenamiento externo de ELS, 47
pruebas de DR con CDRT, 33
pruebas de DR con exportación e importación físicas, 32
pruebas de DR con replicación cruzada de sistemas Tape de VSM, 35
puntos de control
 creación en entornos de VSM, 127
 ejemplos, 128
puntos de control del sistema de recuperación
 creación en entornos de VSM, 127

R

realización de exportaciones e importaciones físicas
 introducción, 37
realización de una verdadera recuperación ante desastres, 29
recuperación ante desastres
 almacenamiento de MVC, 50
 almacenamiento de MVC para LTR, 60
 definición del objetivo de punto de recuperación (RTO), 21
 definición del objetivo de tiempo de recuperación (RTO), 20
 ejecución con la VLE, 134
 introducción, 19
 limpieza con VLE, 138
 manejo de interrupciones temporales, 22
 planificación de pruebas de DR, 30
 pruebas con CTR, 77
 pruebas de DR con CDRT, 33
 pruebas de DR con exportación e importación físicas, 32

pruebas de DR con replicación cruzada de sistemas Tape de VSM, 35
realización de pruebas cuando el sitio de DR no tiene LPAR, 78
realización de una verdadera recuperación ante desastres, 29
recuperación de punto de sincronización, 22
relación de RPO con la recuperación de punto de sincronización, 23
uso de replicación cruzada entre sistemas TapePlex en una solución de DR, 63
recuperación de punto de sincronización, 22
relación de RPO con la recuperación de punto de sincronización, 23
replicación asincrónica
 implementación con supervisión de trabajos, 96
replicación cruzada entre sistemas TapePlex
 configuración e inicio, 66
 configuración para, 65
Consideraciones de VTV de solo lectura, 64
definición de políticas para, 69
descripción, 63
gestión de VTV replicados mediante CTR, 79
pruebas de recuperación ante desastres, 77
uso cuando el sitio remoto no tiene LPAR, 72
uso para continuidad empresarial, 74
uso para reanudación de las actividades empresariales, 76
replicación sincrónica
 implementación, 95

S

sistema de VTSS en cluster
 ejemplo de configuración y gestión, 153

U

uso de configuraciones de VTSS en cluster, 85
uso de la VLE para recuperación ante desastres, 133
uso de replicación cruzada entre sistemas TapePlex en una solución de DR, 63

V

VLE
 ejecución de una prueba de DR, 134
 limpieza después de una prueba de DR, 138

modo de producción normal, 133
uso para continuidad empresarial, 139
uso para recuperación ante desastres, 133
VTSS en cluster
ejemplos, 141
