Planning Disaster Recovery for Microsoft SQL Server Reporting Services in SharePoint Integrated Mode

Jaime Tarquino

Quick Guide

Microsoft

Microsoft

Planning Disaster Recovery forMicrosoft SQL Server Reporting Services in SharePoint Integrated Mode

Author: Jaime Tarquino

Contributors: Nick Swanson, Craig Guyer

Reviewers: James Wu, Dean Kalanquin, Lukasz Pawlowski, Andy Wu, Robert Bruckner

Published: November 2012

Applies to:

- SQL Server 2012 Reporting Services in SharePoint 2010 (with SP1) and SharePoint 2013
- SQL Server 2012 Reporting Services in SharePoint 2013
- SQL Server 2008 R2 Reporting Services in SharePoint 2010 (with SP1)

Summary: This white paper discusses disaster recovery options for Microsoft SQL Server Reporting Services solutions configured to use SharePoint integrated mode. This paper extends best practices for Microsoft SharePoint solutions that include both SQL Server Reporting Services and SharePoint Products. This paper also contains procedures, examples, and scripts that you can use to apply these practices to your organization.

Copyright

Contents

This page intentionally left blank

1. Introduction

This document provides a framework for planning disaster recovery of Microsoft SQL Server Reporting Services deployments that are running in SharePoint integrated mode. The information in this document helps you plan and extend a disaster recovery plan modeled on information in the following SharePoint documents.

- Plan for disaster recovery (SharePoint Server 2010)(http://technet.microsoft.com/library/ff628971).
- Plan for backup and recovery in SharePoint 2013(http://technet.microsoft.com/en-us/library/cc261687(office.15).aspx)

This document discusses the key files and datasets that are important for Reporting Services2008 R2 and Reporting Services2012. The document describes the general actions you to take and the tools you use but the document does not list detailed steps for each combination of versions.

The concepts explained here apply to the following:

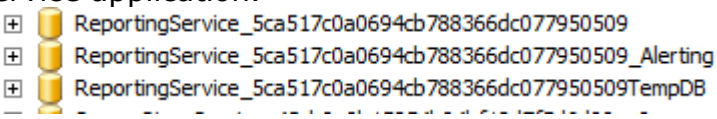
- SQL Server 2012 with Service Pack 1 (SP1)
- SQL Server 2012
- SQL Server 2008 R2

1.1. Assumptions

If you are not familiar with disaster recovery concepts, see the introduction to the white paper Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery (http://msdn.microsoft.com/library/hh781257.aspx).

1.2. Terminology

Database volatility: Database volatilityrefers to how frequent the data is likely to change in the database. In general, the more volatile a database, the more frequent you synchronize the data between a primary and secondary database.
Report server catalog:SQL Server Reporting Services creates the following databases: • ReportServer (**RSDB**) • ReportServerTempDB (**RSTempDB**) • Alerting (only in SQL Server 2012 Reporting Services) Together, the set of databases arecommonly referred to as the Report Server Catalog. Starting with the 2012 version, Reporting Services createsa set of the report server databases for each SSRS service application. These databasesneed to be part of a disaster recovery plan. The following image is shows the default databases created for a 2012

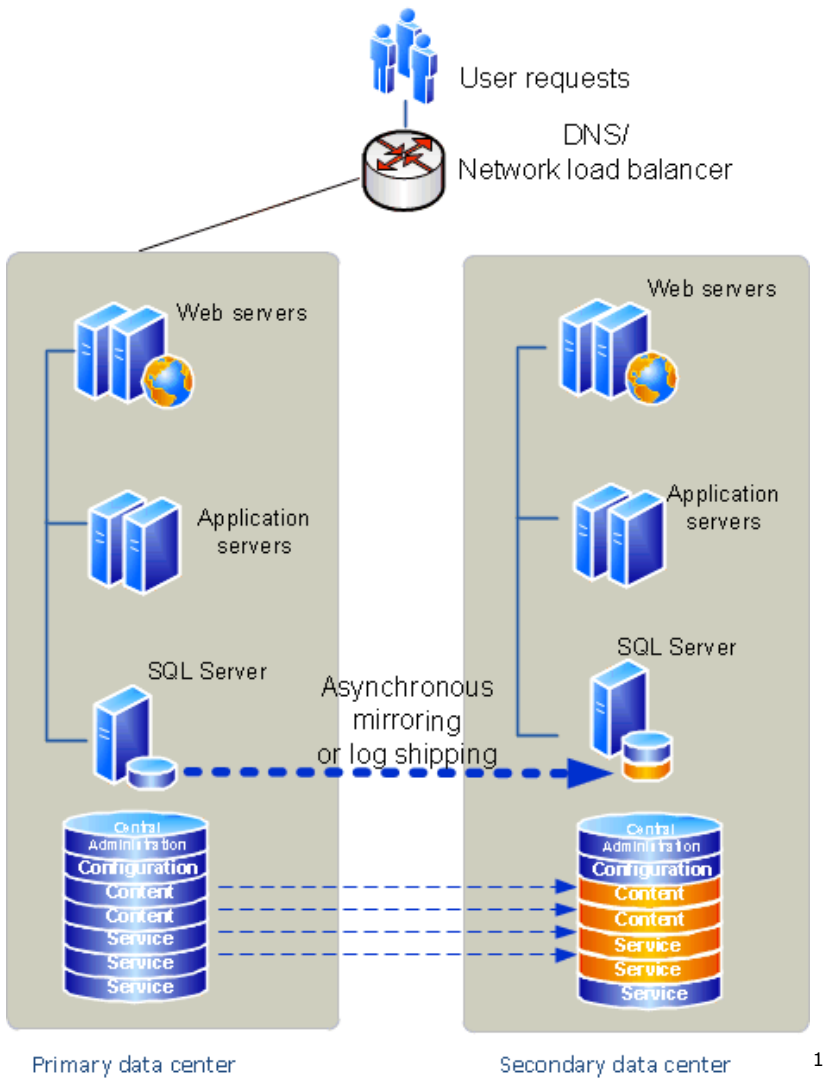
SSRS service application. 
DR site: Inthis paper,"DR" and "DR site" are abbreviations for "Disaster recovery site" or the alternate site you use if your primary site fails.
RS Server: In this paper, "RS Server" is an abbreviation for Reporting Services Serverin SharePoint mode.

2. Overview of SharePoint disaster recovery

The following SharePoint article discusses a basic topology for two datacenters, that are intended toprovide disaster recovery. This topology is also the base topology for the SQL Server Reporting Services plans this paper discusses.

Plan for disaster recovery (SharePoint Server 2010)(http://technet.microsoft.com/en-us/library/ff628971(office.14).aspx).

The SharePoint content databases replicate between the two data centersby either mirroring or log shipping. This replication provides a consistent copy of users' content in case the primary data center is not available.



[1]

[1] Figure from Plan for disaster recovery (SharePoint Server 2010) (http://technet.microsoft.com/en-us/library/ff628971)

3. The Reporting Services Components to Consider for Disaster Recovery

When you design a disaster recovery plan, it is important to identify the different components used by Reporting Services and the relationships with external components.

Consider the following topics when youimplement a disaster recovery solution for SQL Server Reporting Services:

- Reporting Services databases (catalog)
- SharePoint synchronization
- SQL Server Agent
- Scale-out of Report Servers

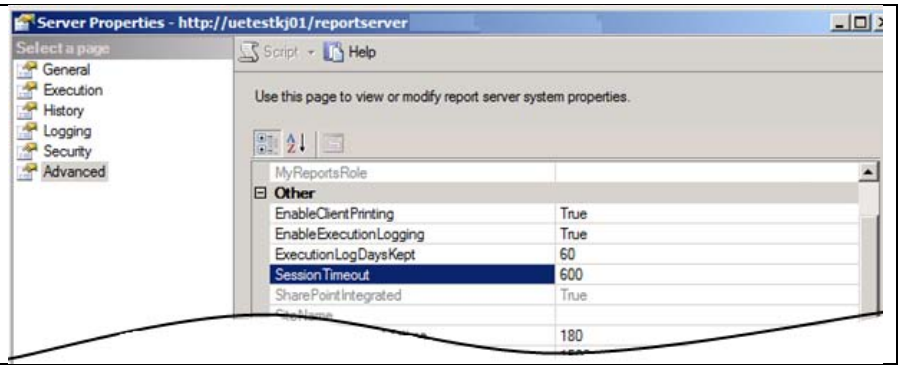
3.1. Reporting Services databases (catalog)

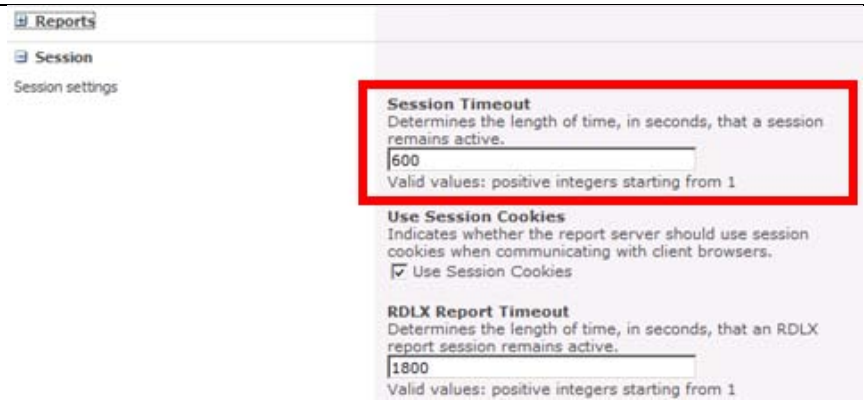
The databases used by Reporting Services can have different disaster recovery plans based on what is best for your environment. The database names in the following list are usedfor reference in the remainder of this document. You can configure a different name for the databases during installation.

RSDB database: This database stores all the metadata, configuration, and definition of every Reporting Services item along with the report execution history and statistics. The volatility of the metadata is typically low; it changes only when items are added or modified. For example, a report design is modified or data source properties are updated. However, the statistics and history change with every execution.

RSTempDB database: This database stores the session information, execution data, cached reports, and work tables that are used during report processing. This database is highly volatile. The minimum lifespan of the data is equal to the SessionTimeout value configured for Reporting Services. The default value for SessionTimeout is 600 seconds (10 minutes).

-

In 2008 R2 Reporting Services, configure SessionTimeout from SQL Server Management Studio. • Right-click the name of the report server, then click **Properties**.	

• Click the **Advanced tab**.	
In 2012 Reporting Services, configure SessionTimeout from SharePoint Central Administration, for each SSRS service application. • Click Manage Service Applications. • Click the name of the SSRS service application then click **System Settings**.	

Alerting database: This database stores the definition, metadata, and runtime of the data alerts feature introduced in SQL Server 2012[2]. The volatility of this database is associated with the minimum time span of the alerts created by users.

3.2. Reporting Services data and SharePoint synchronization

When SQL Server Reporting Services runs in SharePoint mode, it stores the report definition, data sources, models, and other items in SharePoint content databases.Reporting Services also stores a copy of the items in the Report Server Catalog with. The item is stored with information such as connection string information, credentials,and default parameter values.

Reporting Services maintains two copies of report item definitions[3]. One copy is saved in the SharePoint content database and one in the Report Server Catalog (2008 R2) or service application database (2012). Reporting Services manages the synchronization between the two databases with a simple policy for conflicts. The policy checks ifthe item has been modified in SharePoint and updates the version in the Report Server Catalog. TheSharePoint version always takes precedence and the comparison is based on time stamp information. Synchronization is triggered when the item is requested. Requests are created when reports are rendered through a

[2] For more information, see Data Alerts (http://msdn.microsoft.com/library/gg492252(v=SQL.110).aspx).
[3]Except for Reporting Services in local mode for more information, see Reporting Services Report Server (SharePoint Mode)
(http://msdn.microsoft.com/library/hh213532.aspx)

user request or a subscription.Data sources used by a report item are also verified and synchronized.

For more information on the sync feature, see Activate the Report Server File Sync Feature in SharePoint Central Administration (http://technet.microsoft.com/en-us/library/ff487862.aspx).

Because of the synchronization process, it is important your DR plan maintainsparity of the databases between the DR site and primary site. This parity includes the Reporting Services databases and SharePoint content database.

> For example, assumethat the Reporting Services databases on the DR site contain more content than the SharePoint content database on the DR site. When you fail over to the DR site, the reporting services content synchronization process detects extra content in the Report Server catalog because the SharePoint content database takes precedence. Content in the report server catalog is deleted.

It is also important to note that the SharePoint content database does notcontain all the data for the Reporting Services items.The following list shows the data stored only in the Report Server Catalog (service application database):

- Schedules
- Subscriptions
- Cache refresh plans
- Snapshots (report history/execution)
- Credentials for data sources (including embedded)
- Model item security

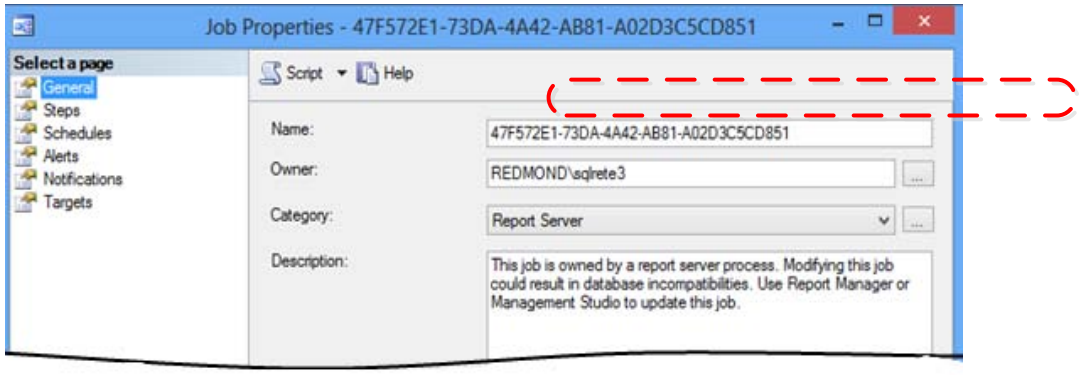
For more information, see Storing and Synchronizing Report Server Content With SharePoint Databases (http://msdn.microsoft.com/library/bb283153(v=sql.105).aspx).

3.3. Reporting Servicesand SQL Server Agent

SQL Server Reporting Services uses SQL Server Agent jobs to execute subscriptions and data alerts atuser-specified dates and times.

Basically,forevery schedule that the user creates (shared or custom) a SQL Server Agent job is created.The job calls a stored procedure to add an event to a table that the SQL Server Reporting Services background process polls every few seconds to start subscription processing.When a new row is found, the server renders the report and attempts to send it to the delivery extension specified by the user. The most common destinations are email, file shares,and SharePoint libraries.

Reporting Services takes care of the creation and updatingof Jobs. A SQL Server Agent job is updatedwhen the schedule is created or modified. Each job created by Reporting Services uses a GUID as a name in the default .NET format of 32 hexadecimal digits separated by hyphens in five groups. The category is set to ReportServer.The following is an example.



3.4. Overview of Reporting Services scale-out deployments

A Reporting Services "report server" is the component thataccepts requests from the client. The server processes the request along with data from the Reporting Services database catalog to render a report and send the result back to the client.

Scaling out Report servers is a key part of a complete disaster recovery plan for Reporting Services.

Every RS Server also executes *background processing*, which handles the following list of tasks:

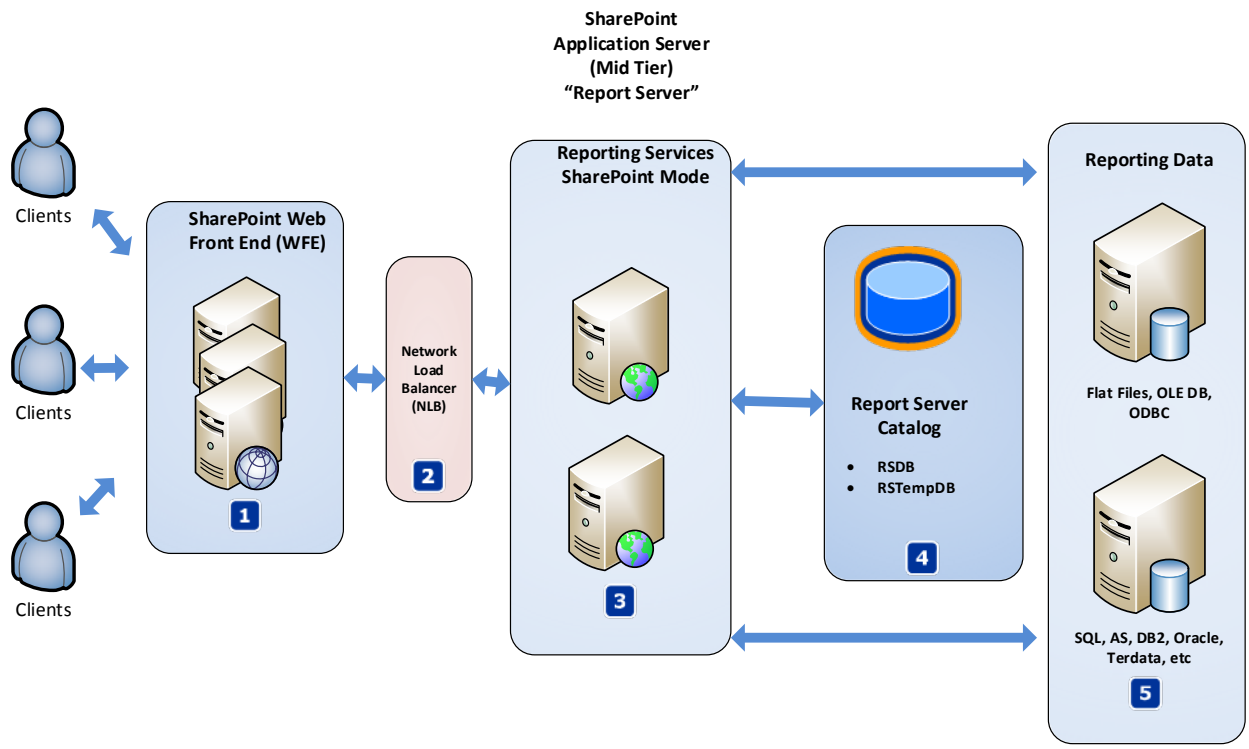
- Process data generated by SQL Server Agent jobs
- Deliver subscription (email, file share, and so on).
- Update cache refresh plans.

- Update snapshots.
- Cleanup the RS Server catalog.

The architecture of the Reporting Services Web Server in SharePoint mode varies depending on the version.Consequently,each version has a different design and tools to implement and maintain scale-out mode.

3.4.1.SQL Server 2008 R2 Reporting Services scale-out architecture

The following diagram illustrates the components of the scale-out architecture of **2008 R2**report servers. The servers are (3) in the diagram.



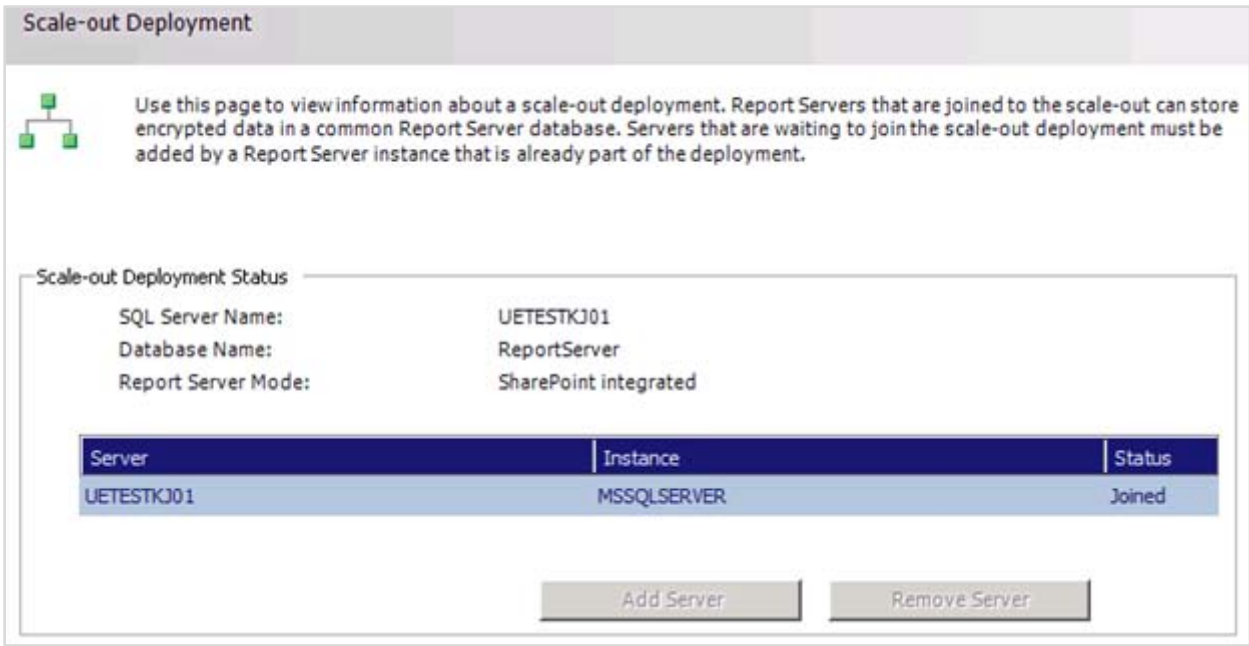
(1)	SharePoint Web Front-end (WFE), each has the Reporting Services add-in for SharePoint products installed (**Rssharepoint.msi**).
(2)	A hardware or software (NLB) network load balancer.
(3)	Multiple report servers that listen to the client requests behind a network load balancer (NLB).
(4)	Each report server (3), is connected to the same RSDB server, and each server can provide reports to the client.

The SharePoint farm is configured as detailed in [How to: Configure Report Server Integration in SharePoint Central]

Administration(http://msdn.microsoft.com/library/bb326213(v=sql.105).aspx).When you implement this scale-out scenario,use the NLB URL address as the Report Server Web Service URL.

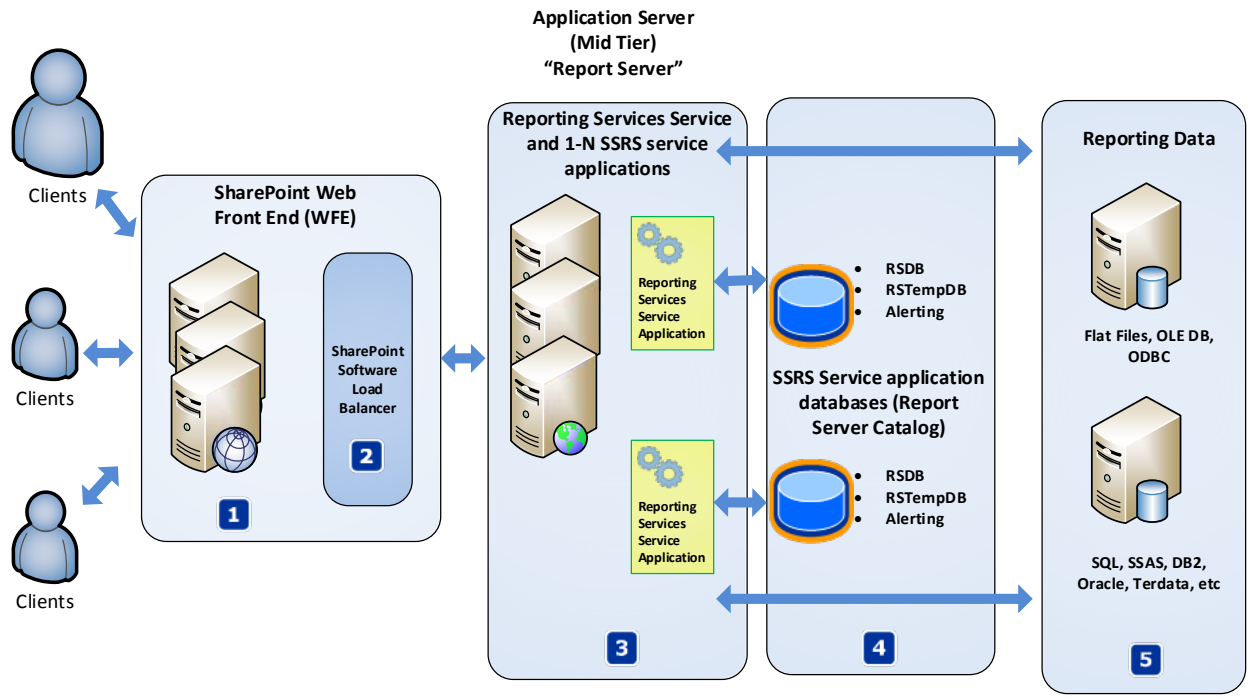
For more information about scale-out architecture, seethe "What is a report server scale-out deployment" section of Deployment Topologies for Reporting Services in SharePoint Integrated Mode (http://msdn.microsoft.com/en-us/library/bb510781(v=SQL.105).aspx).

Reporting Services 2008 R2 scale-out is managed in the Reporting Services Configuration Manager.



3.4.2.SQL Server 2012 Reporting Services scale-out architecture

For SQL Server **2012**Reporting Services, the SharePoint integration has a new architecture as a SharePoint shared service.The main advantage of this architecture from the scale-out point of view is that the shared service uses the SharePoint Software Load Balancer. Therefore, an external NLB component is not required for the RS Server.



(1)	SharePoint Web Front-end (WFE), each has the Reporting Services add-in for SharePoint products installed (**Rssharepoint.msi**).
(2)	Load balancing managed by SharePoint.
(3)	One or more SSRS service applications in the mid-tier of 1-N SharePoint application servers.
(4)	Each SSRS service application has its own set of service application databases.

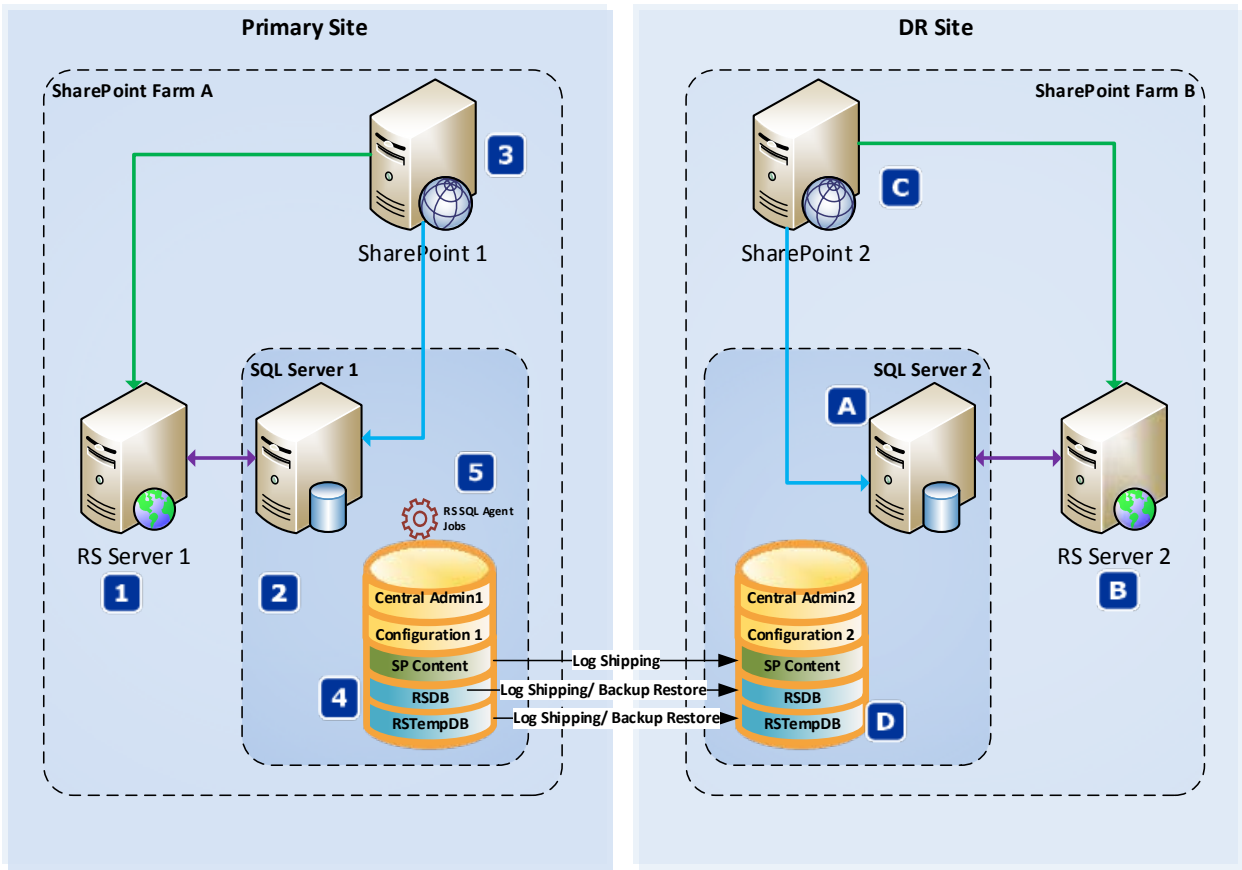
Manage Reporting Services 2012 scale-out in SharePoint Central Administration. You add application servers to the farm and SharePoint managers scale and load balancing.

4. Planning disaster recovery for SQL Server 2008R2Reporting Services

This section describes a general design for disaster recovery for 2008 R2 Reporting Services. The section also describes the general steps and tools you need as part of the plan.

The following figure illustrates a basic DR scenario with two sites:

- The **primary site** handles daily operations.
- Adisaster recovery site (**DR Site**) used when the primary site fails.



Every site has a SharePoint 2010 farm configured as detailed in the article:

[Plan for disaster recovery (SharePoint Server 2010)](http://technet.microsoft.com/library/ff628971)
(http://technet.microsoft.com/library/ff628971).

Every SharePoint farm has aCentral Administration and Configuration Database.The SharePoint Content Database is synchronized with the DR site through log shipping or mirroring, as specified in the article:

Plan for disaster recovery (SharePoint Server 2010)(http://technet.microsoft.com/en-us/library/ff628971).

Each SharePoint farm is configured to its own RS Server, which is in the same site.( and ).

Configure the two RS Servers independently, with each connected to their own database.

4.1. Components of Disaster Recovery for SSRS 2008 R2

To maintain an equivalent environment, the following Reporting Services items need to be the same on both the primary and DR installations.

- Encryption key

  Perform a backup of the encryption key and store the file in a safe location. The key is later restoredin the DR site. For more information, see Restore Encryption Key (Reporting Services Configuration) (http://msdn.microsoft.com/library/bb934307(v=sql.105).aspx).

- Configuration files
  - Rsreportserver.config
  - Rssvrpolicy.config
  - Reportingservicesservice.exe.config
  - Web.config for both the Report Server and Report Manager ASP.NET applications

  Perform the same modifications of the configuration files in the Primary and DR sites. For more information, see Configuration Files (Reporting Services) (http://msdn.microsoft.com/library/ms155866(v=sql.105).aspx).

The common location of the files is the following:

C:\Program Files\Microsoft SQL Server\MSRS10_50.MSSQLSERVER\Reporting Services\ReportServer

- Machine.config for ASP.NETCustom assemblies

Deploy the same set of custom assemblies in the Primary and DR sites. For more information, see Using Custom Assemblies with Reports (http://msdn.microsoft.com/library/ms153561(v=sql.105).aspx).
Unattended execution account, which is configured in the Reporting Services configuration Manager.
• Email configuration in the Reporting Services configuration Manager. For more information, see E-Mail Delivery in Reporting Services (http://msdn.microsoft.com/en-us/library/ms160334(SQL.105).aspx).

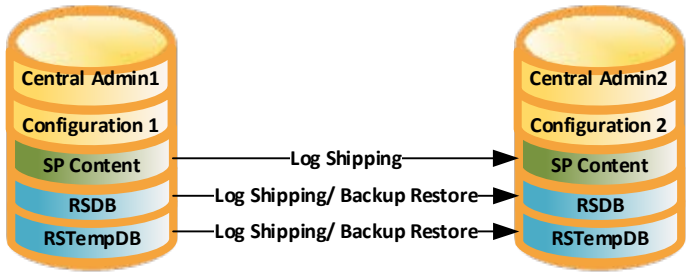
4.2. Report Server catalog

Design data synchronization of the Report Server catalog using one of the following approaches:

- Log shipping.
- Database mirroring.
- Backup and restore of the databases.

The recommendation is that both databases (RSDB and RSTempDB) are replicated with the minimum delta possible between the primary and DR site. It is recommended the sync of the primary and secondary databases be the same for RSDB and RSTempDB. However, you could implementdifferent option for each database based on the volatility of the data.

For example, assume the volume of data changes in the **RSTempDB**and the replication of the databasecreate performance problems. You could implement a less frequent replication than you use for the **RSDB**.



ReportServer database (RSDB): The RSDB metadata changes only when Reporting Services items are modified, for example, when the definition of a report changes, or when its default parameters change. In most situations,

the modification of report items can be considered business data with low volatility. If reports are not frequently created or edited on a frequent basis,then the frequency that RSDB is synchronized should be based on the statistics and execution data.

The statistics and execution history changes with every operation (like rendering a report or executing a subscription); this data has high volatility.If your environment has frequent report executions and you want to ensure that the statistics are not lost, then synchronize RSDB often.

For more information on the execution log, see:

Report Server Execution Log and the ExecutionLog3 View (http://technet.microsoft.com/en-us/library/ms159110.aspx).

RSTempDB: The RSTempDB synchronization window of time can be more flexible,because thedatabase can be re-created at any moment, and it does not need to be prepopulated with data.However,it is important to keep a backup of the database or a copy of the definition script every time you apply anupdateto keep the structure in sync with the RSDB. For example, you install a service pack or a cumulative update on the RS Server.

4.3. Background processing

The Reporting Services service should be stopped on the DR site to avoid background processing on the **[RS Server 2]** server. The databases on the DR site should stay on and synchronized. However, the reporting service should not be started until the DR site is needed and therefore the **[RS Server 2]** server is needed**.** Use the Reporting Servicesconfiguration manager tostop and start Reporting Services.

4.4. Database security roles

In preparation for a failover, the Reporting Services database security role should be provisioned on**[SQL Server 2]** in the DR site.For more information, see:

Create the RSExecRole (http://msdn.microsoft.com/library/cc281308.aspx).

4.5. An example failover cycle for 2008 R2 Reporting Services

If the primary site fails, complete the following failover steps as part of a disaster recovery plan:

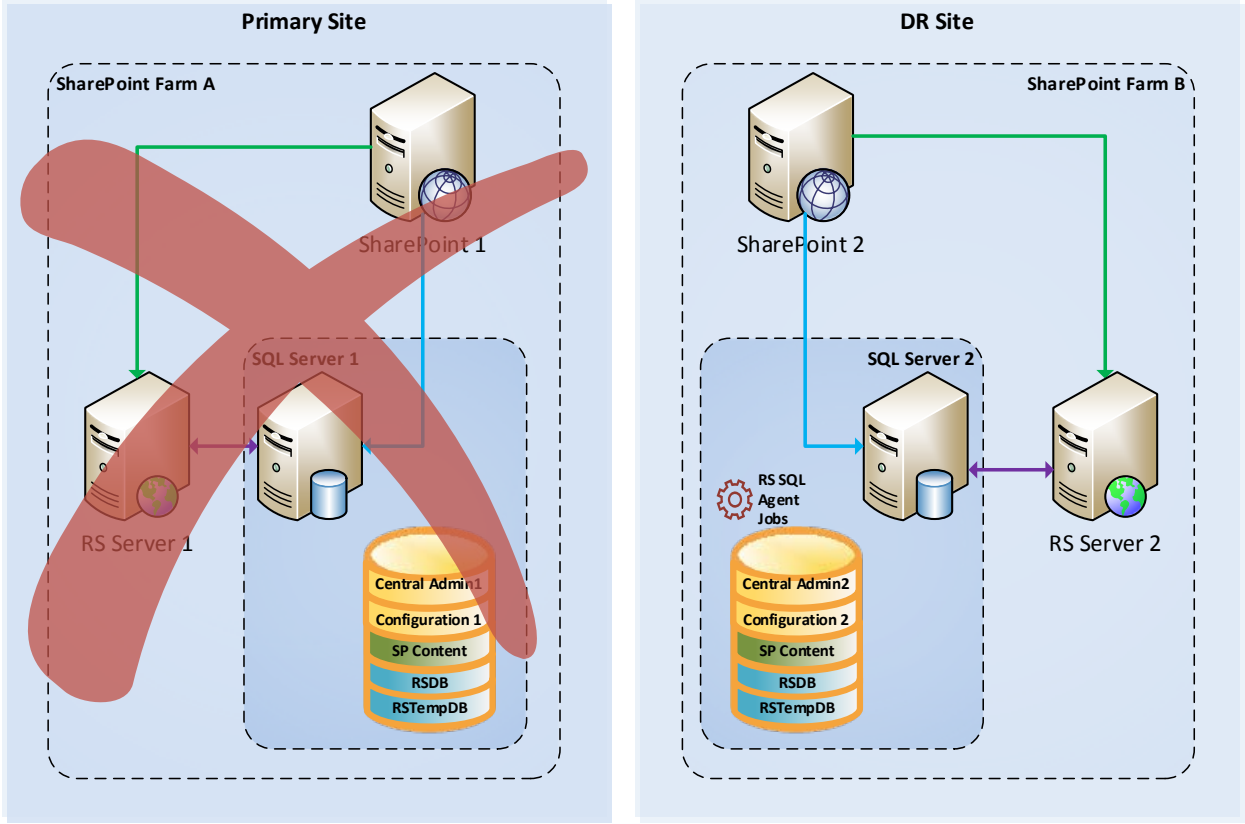
(1)Verify that you have thelatest version of the RS Server catalog in the DR Site, along with the SharePoint content DB. If you need to apply the most recent version of the databases, the process depends on the chosen approach for synchronization in your DR plan.

a. If you are using the backup and restore method, restore the last copy of the database.
b. If you are using log shipping, restore the last backup of the transaction log.
c. If you are using mirroring, execute the mirroring failover procedure.

For more information on how to restore a SQL Server database instance, see. Restore a Database Backup (SQL Server Management Studio) (http://msdn.microsoft.com/en-us/library/ms177429.aspx.

For more information on SharePoint recovery, see the following:

- Plan for backup and recovery in SharePoint Server 2010 (http://technet.microsoft.com/en-us/library/cc261687(office.14).aspx).
- High availability and disaster recovery for SharePoint Server 2010 (http://technet.microsoft.com/en-us/sharepoint/ff601831.aspx).

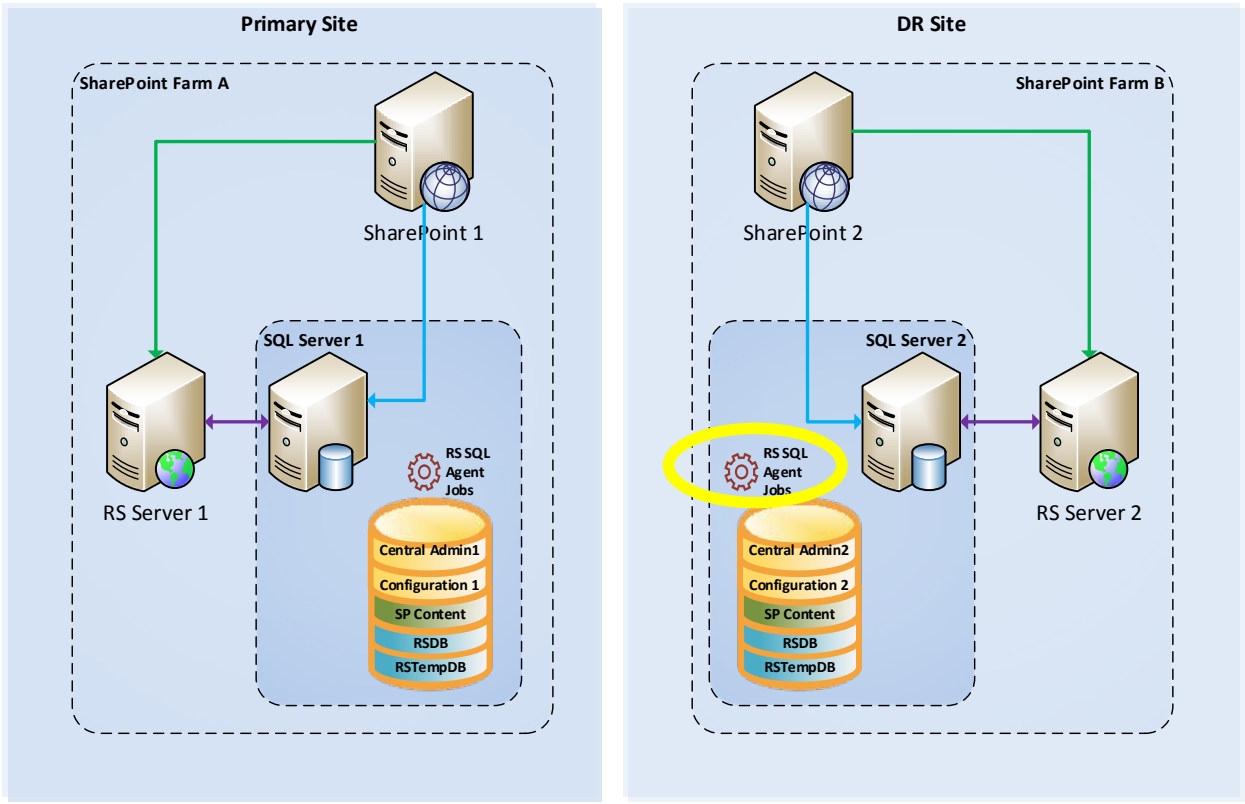


(2) Use the Reporting Services configuration tool, and start the service on the DR site, **[RS Server 2]**.

(3) When **[RS Server 2]** starts, it checks for the existence of SQL Server Agent jobs for its schedule items in RSDB. Because this is a different instance of SQL Server with a different MSDB system database, the jobs do not exist, so RS Server 2 creates the SQL Server Agent jobs.

4.6. Failback

In most cases, when the primary site is back online, or if the DR Site fails, you want to complete a **failback** process or plan. Perform the steps in the failover but in the opposite order. For example, verify and restore the latest backup from the DR site to the primary site. Stop the service on the DR site and start the service on the primary site.



There are special considerations for SSRS SQL Server Agent jobs, because each site (primary and DR) contain the same copies of the SQL Agent jobs. The consequences of are:

- If the database is not available because it is in recovery, (that is, if you are using mirroring or log shipping), the jobs fail and the MSDB job history is "dirty" with recurrent errors.
- If the database is available (that is, if you are using the backup and restore method) the jobs execute and fill the RS catalog with events that the RS Server does not process.
- If a scheduled item (for example, a subscription) is deleted in the current active site, the SQL Server Agent job in charge of that schedule is not deleted in the DR site even after a failover to the DR site. If this job runs, the following message is added the RS Server log:

  (schedule!WindowsService_11!162c!08/08/2012-09:52:50:: w WARN: An event schedule fired that does not exists in the report server database)

In order to minimize the issues, delete any Reporting Services SQL Server Agent jobs on the inactive site. The appendix contains an example of a script that you can use to delete SQL Server Agent jobs.

4.7. Known limitations

- Job execution history is lost between failovers because some of the data is stored in **MSDB**. MSDB is a system database and it cannot be replicated by any of the mentioned methods.
- SQL Server Agent jobs persist between failovers in the inactive site.
- For AlwaysOn, ensure to have only one farm has the RS Service running. Multiple RS Services running in different farms using the same database could lead to unexpected problems.

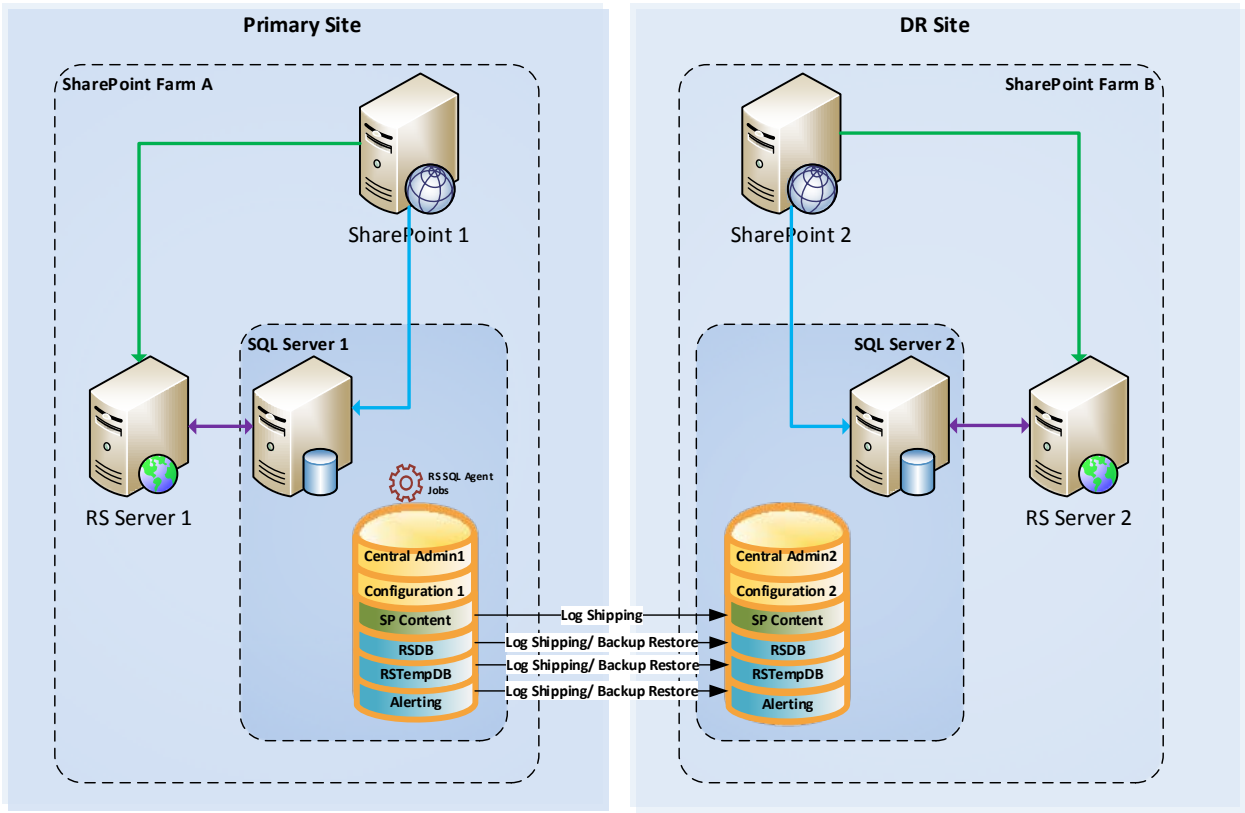
5. Planning disaster recovery for SQL Server 2012 Reporting Services

This section describes a general design of disaster recovery for Reporting Services. The section also describes and the general steps you need to complete for prepare as well as the tools to use for completing the steps.

Most of the options and procedures are the same for SQL Server 2012 Reporting Services as the previous sectionsfor SQL Server 2008 R2 Reporting Services. To avoid confusion, this section contains a modified version of the content from the 2008 R2 section of this document.

The following figure shows a basic scenario with two sites:

- The **primary site**, which handles daily operations.
- A disaster recovery site (**DR Site**), to be usedwhen the primary site fails.



Every site has a SharePoint farm configured as detailed in [Plan for disaster recovery (SharePoint Server 2010)].

Every SharePoint farm has independent Central Administration and Configuration Database. The SharePoint Content Database is synchronizedwith the DR site

through log shipping or mirroring as specified in the articlePlan for disaster recovery (SharePoint Server 2010) (http://technet.microsoft.com/library/ff628971).

Each SharePoint farm is configured to its own RS shared service, which is in the same farm.

The two Reporting Servicesservers are configured to use the same Report Server catalog.

5.1. Components of Disaster Recovery for SSRS 2012

To keep an equivalent environment, replicate the following Reporting Services items between the two sites:

- Encryption key

Perform a backup of the encryption key and store the file in a safe location. The key is restoredlater in the DR site. For more information, see the section Key Management in Manage a Reporting Services Service Application (http://technet.microsoft.com/library/gg492284(SQL.110).aspx).

- Configuration files
  - Rsreportserver.config
  - Rssvrpolicy.config

Perform the same modifications of the configuration files in the Primary and DR sites. For more information, see Configuration Files (Reporting Services) (http://technet.microsoft.com/library/ms155866.aspx).

The common location of the files is the following:

C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\WebServices\Reporting

- Machine.config for ASP.NETCustom assemblies

Deploy the same set of custom assemblies in the Primary and DR sites. For more information, see Using Custom Assemblies with Reports (http://msdn.microsoft.com/library/ms153561(v=sql.110).aspx).

- Unattended execution account, which is configured in SharePoint Central Administration.

  Configuration of the Reporting Services service application is managed from SharePoint Central Administration. For more information, see Manage a Reporting Services Service Application (http://technet.microsoft.com/library/gg492284(SQL.110).aspx).

- Email configuration for each SSRS service application, Managed in SharePoint Central Administration.

  See "E-mail settings" section of Manage a Reporting Services Service Application (http://technet.microsoft.com/library/gg492284(SQL.110).aspx).

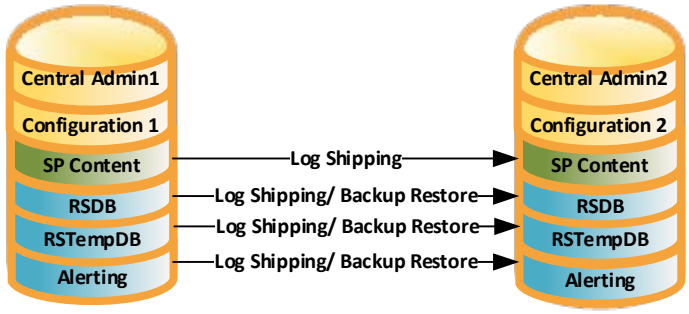
5.2. Report Server catalog

You can design data synchronization of the Report Server catalog using one of the following approaches:

- Log shipping.
- Database mirroring
- Backup and restore of the databases.

The recommendation is that both databases (RSDB and RSTempDB) are replicated with minimaldelta between the primary and DR site. It is recommended the sync of the primary and secondary databases be the same for RSDB and RSTempDB. However, you could implement different option for each database based on the volatility of the data.

For example, assume the volume of data changes in the **RSTempDB** and the replication of the databasecreate performance problems. You could implement a less frequent replication than you use for the **RSDB**.



ReportServer database (RSDB): The RSDB metadata changes only when Reporting Services items are modified. For example, when the definition of a report changes, or when the reports default parameters change. In most situations, the modification of report items can be considered business data with low volatility. If reports are not created or edited on a frequent basis then the frequency that RSDB is synchronized, should be based on the statistics and execution data.

The statistics and execution history changes with every operation (like rendering a report or executing a subscription); this data has high volatility.If your environment has many report executions and you want to ensure that the statistics are not lost, then synchronize RSDB often.

For more information on the execution log, see:

Report Server Execution Log and the ExecutionLog3 View (http://technet.microsoft.com/en-us/library/ms159110.aspx).

RSTempDB: The RSTempDB synchronization time window can be more flexible, because this database can be re-created at any moment, and it does not need to be prepopulated with data. However, it is important to keep a backup of the database or a copy of the definition script every time you apply an update to the RS Server, to keep the structure in sync with the RSDB. For example, whenever you apply a service pack or a cumulative update.

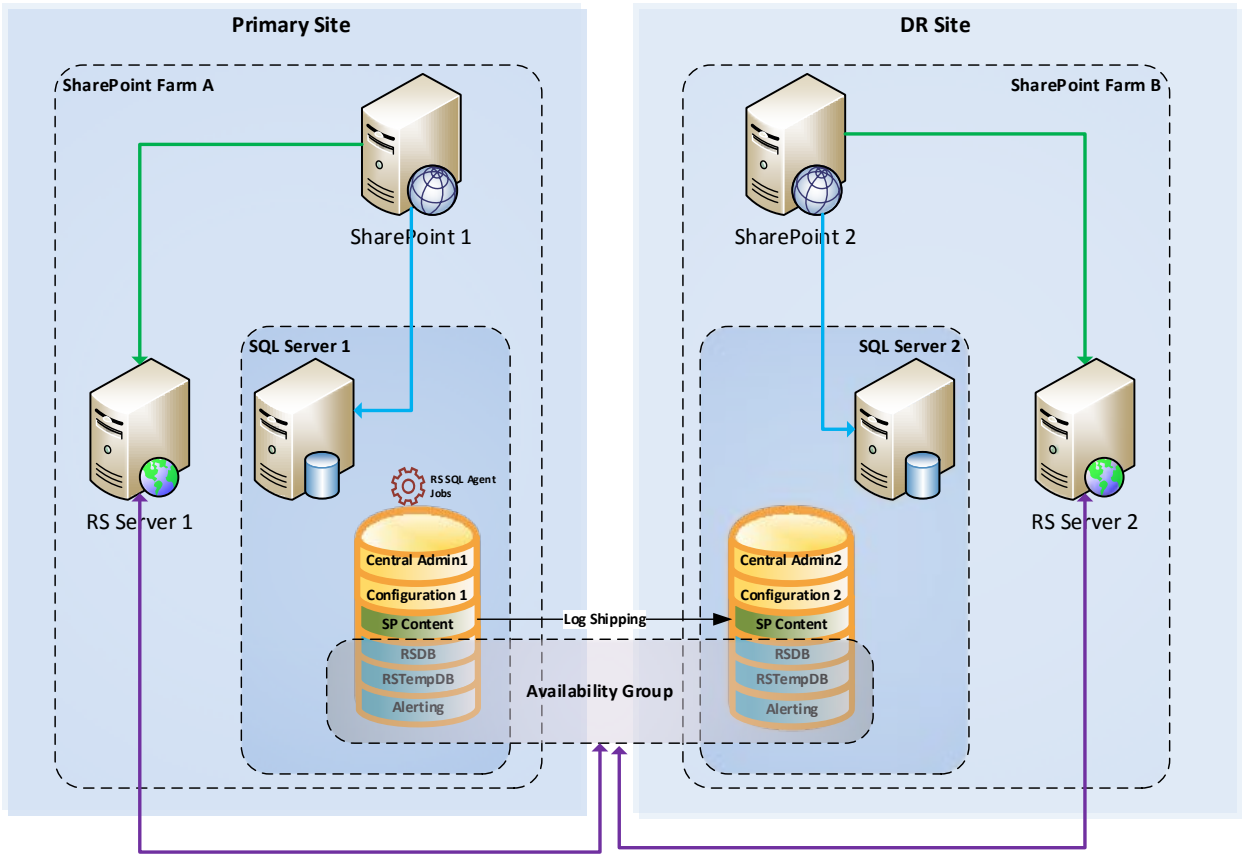
Alerting: The synchronization window for the Alerting database should be the same as the RSDB synchronization window. There is a strong dependency between the alerts and the reports which that the alerts are based upon.

Introduced in SQL Server 2012,AlwaysOn is a new way to provide disaster recovery for the Report Server catalog.To provide consistency across the Reporting Services databases,the Report Server catalog database should be in the same *availability group*.

For more information about availability groups, see the following:

- Failover Modes (AlwaysOn Availability Groups) (http://msdn.microsoft.com/library/hh213151(SQL.110).aspx).

The following diagram illustrates the DR plan usingAlwaysOn.



Regardless of the failover mode configured for AlwaysOn, it offersthe following advantages:

- The availability group reduces the administrative effort to bring the DR site online, because the recovery is the group rather than three independent databases.

For more information, see the following:

- Reporting Services with AlwaysOn Availability Groups (SQL Server) (http://msdn.microsoft.com/en-us/library/hh882437.aspx).
- Microsoft SQL Server AlwaysOn Solutions Guide for High Availability and Disaster Recovery (http://msdn.microsoft.com/library/hh781257.aspx).

5.3. Background processing

The Reporting Services service on the DR site **[RS Server2]** should not be running. This is to avoid background processing on the **[RS Server 2]**server. The databases used on the DR site should stay active and synchronized but the Reporting Services service should be stopped until the DR site is needed. Use SharePoint Central

administration to Stop and Start the Reporting Services service.For more information, see:

Manage services on the server (SharePoint Server 2010) (http://technet.microsoft.com/library/ee704549.aspx).

5.4. Database security roles

In preparation for a failover, the Reporting Services database security role should be provisioned in **[SQL Server 2]**on the DR site.For more information about security roles, see:

Create the RSExecRole (http://msdn.microsoft.com/library/cc281308.aspx).

5.5. An Example Failover Cycle for 2012 Reporting Services

If the primary site fails, complete the following failover steps as part of a disaster recovery plan:

If you are Not using AlwaysOn

1. **Verify**that you have the latest version of the RS Server catalog in the DR Site, along with the SharePoint content DB. If you need to apply the most recent version of the databases, the process depends on what you chose for synchronization in your DR plan.
    a. If you are using the backup and restore method, restore the last copy of the database.
    b. If you are using log shipping, restore the last backup of the transaction log.
    c. If you are using mirroring, execute the mirroring failover procedure.
2. To start the service, use SharePoint Central Administration in the DR site.

If you are Using AlwaysOn

1. To make the database available on the DR site, use automatic or manual failover.
2. To start the service, use SharePoint Central Administration in the DR site.
3. Ensure that the RS Service in the Primary Site is stopped.

Tip: After you start the RS Service in Central Administration, reports cannot render until the next execution of the Application Addresses Refresh Job,which registers the service address. By default it runs every 15 minutes. You can reduce the time by executing the job from SharePoint Central Administration, for more information about the timer jobs, see Timer job reference (SharePoint Foundation 2010) (http://technet.microsoft.com/library/ff808317.aspx). Repeat the same steps for any additional RS shared service application configured in your farm.

When the **[RS Server 2]** starts, it checks for the existence of the SQL Server Agent jobs needed to schedule base operations. Because it is a different instance of SQL Server with a different MSDB system database, the jobs do not exist.

Therefore the **[RS Server 2]** creates the jobs.

5.6. Failback

In most cases, when the primary site is back online, or if the DR Site fails, you want to complete a **failback** process or plan. Perform the steps in the failover but in the opposite order. For example, verify and restore the latest backup from the DR site to the primary site. Stop the service on the DR site and start the service on the primary site.

There are special considerations for Reporting Servicesrelated SQL Server Agent jobs, because each site contains the same copies of the jobs. The consequences are:

- If the database is not available because it is in recovery, (that is, if you are using mirroring or log shipping), the jobs fail and the MSDB job history is "dirty" with recurrent errors.
- If the database is available (that is, if you are using the backup and restore method) the jobs execute and fills the SSRS catalog with events that the Reporting Services Server does not process.
- If a schedule item (like a subscription) is deleted in the current active site, the SQL Server Agent job in charge of that schedule is not deleted in the DR site even after a failover to the DR site. If this job runs, the following message is added the RS Server log:
  (schedule!WindowsService_11!162c!08/08/2012-09:52:50:: w WARN: An event schedule fired that does not exists in the report server database)

To minimize theissues,delete the Reporting Services SQL Server Agent jobs on the inactive site.The appendixcontains an example of a script that you can use to delete SQL Server Agent jobs.

5.7. Known limitations

- Job execution history is lost between failovers because some of the data is stored in **MSDB**. MSDB is a system database and it cannot be replicated by any of the mentioned methods.
- SQL Server Agent jobs persist between failovers in the inactive site.

- For AlwaysOn, ensure only one of the farms RS Service is running. Multiple RS Services running in different farms using the same database could lead to unexpected problems.

6. Conclusion

Reporting Services in SharePoint integrated mode uses multiple products and services to offer an integrated experience. Because of this,consider each component in a disaster recovery plan.The best planfor your situation depends on the specific business requirements.This document can help you identify each component, and it provides the most common solutions to implement disaster recovery for them.

For more information:

Plan for backup and recovery in SharePoint Server 2010 (http://technet.microsoft.com/en-us/library/cc261687(office.14).aspx)
Plan for disaster recovery (SharePoint Server 2010) (http://technet.microsoft.com/library/ff628971), offers a good overview. However, it does not provide in-depth information on how to provide the same level of disaster recovery for Reporting Services, which this document covers.
A brief look at Reporting Services and SQL Server AlwaysOn availability groups can be found in the topic "Reporting Services with AlwaysOn Availability Groups (SQL Server)" (http://msdn.microsoft.com/en-us/library/hh882437.aspx).
Where to find the Reporting Services add-in for SharePoint Products (http://technet.microsoft.com/en-us/library/gg426282.aspx).
SQL Server Web site: http://www.microsoft.com/sqlserver/:
SQL Server TechCenter: .http://technet.microsoft.com/en-us/sqlserver/
SQL Server DevCenter: http://msdn.microsoft.com/en-us/sqlserver/

Did this paper help you? Please provide feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

- Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
- Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback helps us improve the quality of white papers.

Send feedback.(mailto:sqlfback@microsoft.com)

7. Appendix – SQL script to delete SQL Server Agent Jobs

Here is a transact SQL script you can use to delete Reporting Services SQL Server Agent jobs from the inactive site.

```
usemsdb
declare@jobnamesysname,@retint
declaredelete_job_cursorcursorfor

selectdistinctj.name
frommsdb.dbo.sysjobsj
joinmsdb.dbo.sysjobstepsjsonjs.job_id=j.job_id
joinmsdb.dbo.syscategoriesconc.category_id=j.category_id
wherej.namelike'[0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-
9a-fA-F][0-9a-fA-F]-[0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F]-[0-9a-fA-F][0-9a-fA-
F][0-9a-fA-F][0-9a-fA-F]-[0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F]-[0-9a-fA-F][0-9a-
fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-9a-fA-F][0-
9a-fA-F][0-9a-fA-F][0-9a-fA-F]'

opendelete_job_cursor
fetchnextfromdelete_job_cursorinto@jobname
while@@FETCH_STATUS= 0
begin

        print'Deleting jobname = '+@jobname

        exec@ret=sp_delete_job@job_name=@jobname

        if(@@error<> 0 or@ret<> 0)
        begin
                break
        end
        fetchnextfromdelete_job_cursorinto@jobname
end
closedelete_job_cursor
deallocatedelete_job_cursor
```