

Oracle® Solaris Cluster Data Replication Guide for ZFS Snapshots

ORACLE®

Part No: E69466
October 2018

Part No: E69466

Copyright © 2016, 2018, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

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.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Référence: E69466

Copyright © 2016, 2018, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

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.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou ce matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

Contents

Using This Documentation	9
1 Setting Up Oracle Solaris ZFS Snapshot Protection Groups	11
Overview of Data Replication Using Oracle Solaris ZFS Snapshot	11
About Oracle Solaris ZFS Snapshots Replication	14
Oracle Solaris ZFS Snapshot Replication Package	19
Oracle Solaris ZFS Snapshot Infrastructure Resource Groups	19
Oracle Solaris ZFS Snapshot Replication Agent Resource Groups	21
Oracle Solaris ZFS Snapshot Replication Status Resource Groups	23
Replicating Data in an Oracle Solaris ZFS Snapshot Protection Group (Task Maps)	23
Planning Remote Replication Using Oracle Solaris ZFS Snapshot	24
Guidelines for Remote Replication Using Oracle Solaris ZFS Snapshot	25
Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File	26
Disaster Recovery Framework Properties for Oracle Solaris ZFS Snapshot Replication	29
Remote Replication Layer Process for Validating the Application Resource Groups and Remote Replication Entities	30
Creating and Validating an Oracle Solaris ZFS Snapshot Protection Group	31
Strategies for Creating Oracle Solaris ZFS Snapshot Protection Groups	31
Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot	32
▼ How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group	34
Debugging an Oracle Solaris ZFS Snapshot Protection Group	40
Starting an Oracle Solaris ZFS Snapshot Protection Group	42
Stopping an Oracle Solaris ZFS Snapshot Protection Group	43

Adding a Replication Component to an Oracle Solaris ZFS Snapshot Protection Group	44
▼ How to Add a Replication Component to an Oracle Solaris ZFS snapshot Protection Group	44
Adding an Application Resource Group to an Oracle Solaris ZFS snapshot Protection Group	50
▼ How to Add an Application Resource Group to an Oracle Solaris ZFS Snapshot Protection Group	50
2 Administering Oracle Solaris ZFS Snapshot Protection Groups	55
Replicating an Oracle Solaris ZFS Snapshot Protection Group Configuration to a Partner Cluster	55
▼ How to Replicate the Oracle Solaris ZFS Snapshot Protection Group Configuration to a Partner Cluster	56
Administering Oracle Solaris ZFS Snapshot Replication Components	59
▼ How to Modify an Oracle Solaris ZFS Snapshot Replication Component	59
▼ How to Remove a Replication Component From an Oracle Solaris ZFS Snapshot Protection Group	63
Changing Oracle Solaris ZFS Snapshot Application Resource Groups Between Failover and Scalable	66
▼ How to Change the Resource Type of an Application Resource Group	66
Removing Oracle Solaris ZFS Snapshot Application Resource Groups	67
▼ How to Remove an Application Resource Group From an Oracle Solaris ZFS Snapshot Protection Group	67
Deleting Oracle Solaris ZFS Snapshot Protection Group From a Partner Cluster	69
▼ How to Remove a Oracle Solaris ZFS Snapshot Protection Group From a Partner Cluster	70
Checking the Runtime Status of Oracle Solaris ZFS Snapshot Remote Replication	72
Overview of Displaying an Oracle Solaris ZFS Snapshot Runtime Status	72
▼ How to Check the Runtime Status of Oracle Solaris ZFS Snapshot Replication	73
Oracle Solaris ZFS Snapshot Replication Resource Runtime Status and Status Messages	74
3 Migrating Services That Use Oracle Solaris ZFS Snapshot Remote Replication	77
Switchover Services to a Partner Cluster in a Planned Manner	77

▼ How to Perform a Switchover for a Oracle Solaris ZFS Snapshot Protection Group	77
Migrating Services That use ZFS Snapshot Replication with a Takeover	79
▼ How to Perform a Takeover for a Oracle Solaris ZFS Snapshot Protection Group	79
Recovering Services to a Cluster on a System That Uses Oracle Solaris ZFS Snapshot Replication	80
Overview of Recovering Services	80
▼ How to Perform a Failback-Switchover on a System That Uses Oracle Solaris ZFS Snapshot Replication	81
▼ How to Perform a Failback-Takeover on a System That Uses Oracle Solaris ZFS Snapshot Replication	84
Recovering From an Oracle Solaris ZFS snapshot Remote Replication Error	87
▼ How to Detect Remote Replication Errors	87
▼ How to Recover From an Oracle Solaris ZFS snapshot Remote Replication Error	88
A Disaster Recovery Framework Properties for Oracle Solaris ZFS Snapshot	91
Extension Properties for Oracle Solaris ZFS Snapshot Replication Agent Resource	91
Extension Properties for Oracle Solaris ZFS Snapshot Resource Status	95
B Use Cases for Oracle Solaris ZFS Snapshot Replication	99
Configuration Assumptions of Use Cases	99
Use Case: Setting Up Replication User and SSH	100
Use Case: Configuring Oracle Solaris ZFS Snapshot Replication When Both Partner Clusters Are Global Zones	108
Use Case: Configuring Oracle Solaris ZFS Snapshot Replication With One Partner Cluster as Zone Cluster	113
Use Case: Configuring Oracle Solaris ZFS Snapshot Replication With Both Partner Clusters as Zone Cluster	119
Use Case: Configuring Oracle Solaris ZFS Snapshot Replication With Zpools for Globally Mounted ZFS Filesystems	125
C Configuring Oracle Solaris ZFS Snapshot Replication for ZFS-based Cluster Filesystems in Zone Clusters	131
Configuration Requirements	131

Use Case: Configuring Oracle Solaris ZFS Snapshot Replication for ZFS-based Cluster Filesystems in Zone Clusters	132
Index	157

Using This Documentation

- **Overview** – Describes how to use the ZFS snapshot feature of Oracle Solaris for data replication for disaster recovery.
- **Audience** – Experienced system administrators with extensive knowledge of Oracle software and hardware.
- **Required knowledge** – Knowledge of the Oracle Solaris operating system and of Oracle Solaris Cluster software, and expertise with the volume manager software that is used with Oracle Solaris Cluster software.

Product Documentation Library

Documentation and resources for this product and related products are available at http://www.oracle.com/pls/topic/lookup?ctx=E69294_01.

Feedback

Provide feedback about this documentation at <http://www.oracle.com/goto/docfeedback>.

Setting Up Oracle Solaris ZFS Snapshot Protection Groups

This chapter contains information about configuring data replication using Oracle Solaris ZFS snapshot in an Oracle Solaris Cluster disaster recovery framework setup. You can configure periodic asynchronous replication of application data in a disaster recovery framework protection group using ZFS snapshot-based data replication.

This chapter covers the following topics:

- [“Replicating Data in an Oracle Solaris ZFS Snapshot Protection Group \(Task Maps\)” on page 23](#)
- [“Planning Remote Replication Using Oracle Solaris ZFS Snapshot” on page 24](#)
- [“Creating and Validating an Oracle Solaris ZFS Snapshot Protection Group” on page 31](#)
- [“Adding a Replication Component to an Oracle Solaris ZFS Snapshot Protection Group” on page 44](#)
- [“Adding an Application Resource Group to an Oracle Solaris ZFS snapshot Protection Group” on page 50](#)

Overview of Data Replication Using Oracle Solaris ZFS Snapshot

ZFS has built-in features that perform the following actions:

- Take a consistent snapshot of a ZFS dataset.
- Create a stream representation of such a snapshot.
- Create a snapshot based on the contents of such a stream.

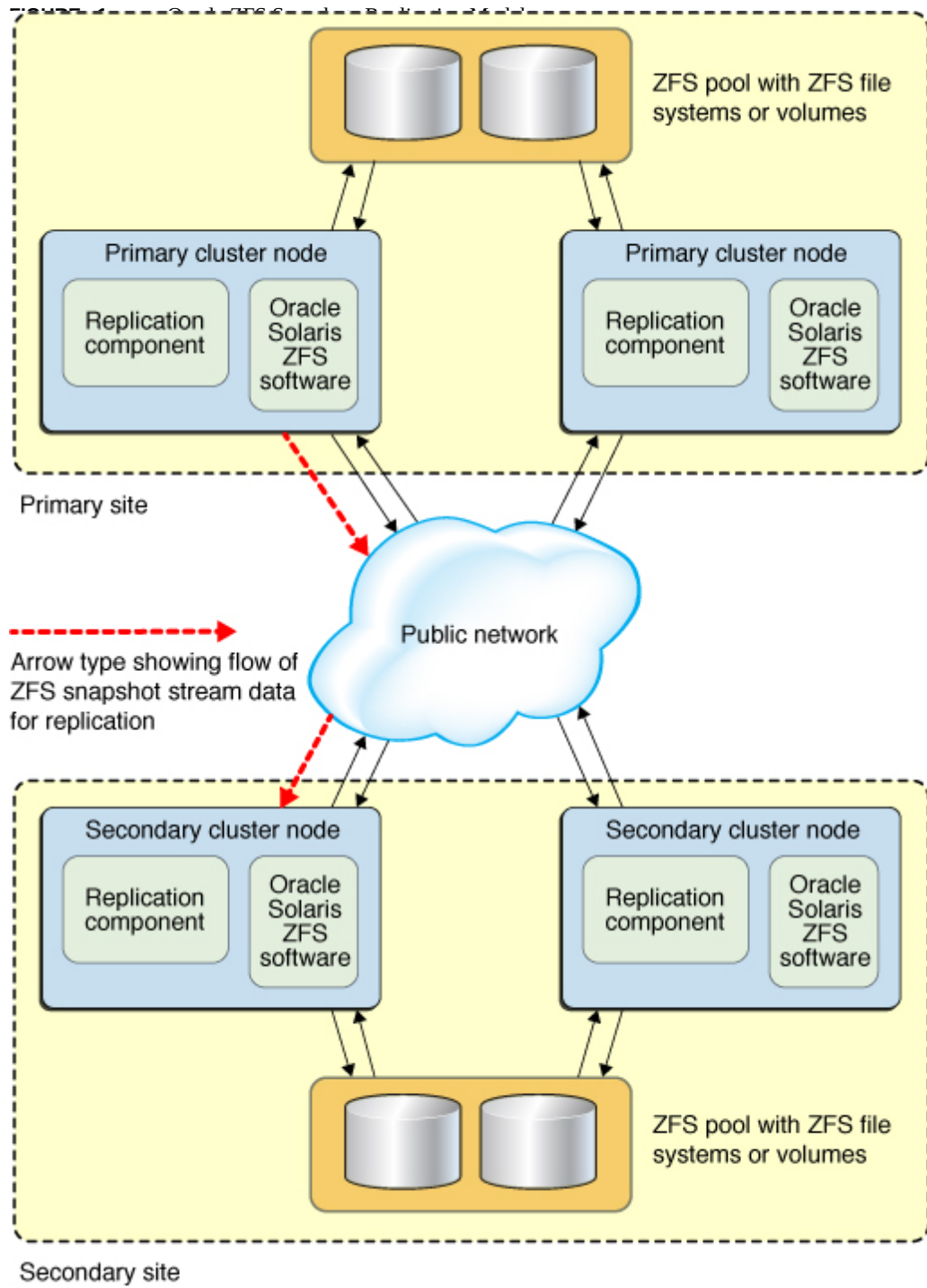
Such features of ZFS can be used together to provide a method of replicating data on ZFS datasets from the primary cluster to the secondary cluster within a DR environment. This process ensures that the contents of the primary and secondary datasets are in sync. This is the

basis of this replication feature supported by disaster recovery framework which in turn controls and monitors the direction of replication.

During data replication, data from a primary cluster is copied to a backup or secondary cluster. The secondary site can be located at a geographically separated location from the primary site.

Before you can replicate data with the Oracle Solaris ZFS snapshots feature, you must be familiar with Oracle Solaris ZFS administration and have the latest Oracle Solaris SRUs installed on your system. For information about Oracle Solaris ZFS, see [Managing ZFS File Systems in Oracle Solaris 11.4](#). For information about replicating data using Oracle Solaris ZFS snapshot, see [Chapter 8, “Working With Oracle Solaris ZFS Snapshots and Clones” in Managing ZFS File Systems in Oracle Solaris 11.4](#). For information about ZFS commands, see the `zfs(8)` man page.

The following figure displays how Oracle Solaris ZFS snapshot is used to replicate data.



The basis of the disaster recovery framework ZFS snapshot replication module is to periodically do this : create a ZFS snapshot on the source dataset, use the `zfs send` command to create a snapshot stream based on the snapshot, and replicate that snapshot on to the target dataset by feeding the stream to `zfs receive` command on the target host over SSH. Data replication is performed with the credentials of a replication user on each cluster.

The replication module consists of the following components:

- HAStoragePlus resource for highly available ZFS pools in the Oracle Solaris Cluster environment.
- Logical host name resource to manage host name for replication source or target.
- Replication user with the credentials to perform snapshots replication.
- Replication user's SSH setup with public/private keys and passphrase. You must be able to establish SSH connection between any node of the primary cluster and any node of the secondary cluster using replication user's SSH credentials.

The figure shows that the replication configuration exists on all nodes. However, functional parts such as the ZFS pool, HAStoragePlus resource, and logical hostname resource are active on one primary node and one secondary node at a time.

You must use the configuration file and the setup script provided with disaster recovery framework ZFS snapshots replication package to set up a protection group with ZFS snapshot replication on the primary cluster in a partnership. For more information about the parameters used in the configuration file, see [“Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File”](#) on page 26.

About Oracle Solaris ZFS Snapshots Replication

The basis of the disaster recovery framework ZFS snapshot replication module is to periodically performs the following actions:

- Create a ZFS snapshot on the source dataset.
- Create a snapshot stream using the `zfs send` command
- Replicate the snapshot on to the target dataset by feeding the stream to the `zfs receive` command on the target host over SSH.

The replication module names the replication snapshots in the format `oscge_num`, where *num* is a monotonically increasing sequence number.

ZFS supports different types of snapshot streams and stream packages. disaster recovery framework ZFS snapshot replication module supports full streams as well as incremental streams. If source and target datasets already have common snapshots thus enabling replication

using incremental streams, then this replication module uses incremental streams. Otherwise, it uses full streams. The replication module always attempt to use incremental replication wherever possible. The replication module takes the decision to use the incremental replication and there is no option available to set this behavior explicitly.

This snapshot replication module supports data replication using ZFS replication stream package or ZFS recursive stream package. You can use these options to replicate a source dataset and its child datasets on to the target cluster. For example, suppose the source dataset is `src-pool1/app1-ds1` and the target dataset is `targ-pool1/app1-ds1-copy`. The source dataset has child datasets `src-pool1/app1-ds1/fs1` and `src-pool1/app1-ds1/vol1`. Then either of these stream package options can be used to replicate the dataset hierarchy starting at `src-pool1/app1-ds1` on to the target. As a result of the replication, the data of source dataset `src-pool1/app1-ds1` is replicated to the target dataset `targ-pool1/app1-ds1-copy`, the data of `src-pool1/app1-ds1/fs1` is replicated to `targ-pool1/app1-ds1-copy/fs1`, and the data of `src-pool1/app1-ds1/vol1` is replicated to `targ-pool1/app1-ds1-copy/vol1`.

For more information about ZFS snapshot streams, see “[Types of ZFS Snapshot Streams](#)” in *Managing ZFS File Systems in Oracle Solaris 11.4*. For more information about “`zfs send`” and “`zfs receive`” commands and their options related to snapshot streams and stream packages, see the [zfs\(8\)](#) man page.

Resumable Replication

New options available with the `zfs send` and `zfs receive` commands enable the replication to resume at the point where the previous transfer was interrupted.

The `zfs receive` command on the target system determines where the transfer was interrupted. The `zfs send` command uses the information gathered by `zfs receive` command on the source system to generate a new send stream that resumes at the point the previous transfer was interrupted. Finally, `zfs receive` will splice the resumed stream together with the previously received data. This process is repeated as many times as necessary until the stream is successfully transferred or a non-resumable error is encountered.

The following new options are available with the `zfs send` and `zfs receive` commands:

- `zfs receive -C` – Writes a receive checkpoint to stdout. A receive checkpoint is a table of contents of snapshots already received
- `zfs send -C` – Reads a receive checkpoint from stdin. A receive checkpoint is a table of contents of snapshots that have already been successfully received and do not need to be sent again.
- `zfs list -I` – Recursively displays incomplete datasets that are in the state specified.

- `zfs send -s nocheck` – Specifies that the output stream uses a legacy send stream format without per-record checksums. This format is suitable for transmission to older systems that do not support the new format.

The replication module finds the common snapshots among all the hierarchies of the source dataset. Among them, the latest snapshot found is used as the snapshot name to send. If no such snapshots exist, a new snapshot is created and sent with `zfs send -C` option. This operation will resume the replication from the point where it was interrupted in the previous attempt. If this operation fails, replication module deletes all the resumables at the target and executes a normal replication with `zfs send` without the `-C` option.

For example, on a system containing a ZFS dataset name `src`, the following command transfers the snapshot to the dataset hierarchy at `dst` on the target system.

```
# zfs send -R src@TOSNAP | ssh target zfs receive -F dst
```

If the above command is interrupted, type the following command to resume the interrupted transfer.

```
# ssh target zfs receive -C dst | zfs send -RC src@TOSNAP | ssh target zfs receive -F dst
```

Type the following command to display all the incomplete datasets.

```
# zfs list -I all
```

```
NAME USED AVAIL REFER TYPE STATE tank/dest/one/%now 521M
120G 521M volume receiving tank/dest/two 33.8M 120G 33.8M filesystem resumable
```

Replication Stream Package

You can generate a replication stream package using the `-R` option with the `zfs send` command. You can choose this type of replication by using the `REPLICATION_STREAM_PACKAGE` parameter in the replication parameters configuration file. For more information about the parameters available in the configuration file, see [“Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File” on page 26](#).

You can use replication stream package in one of the following situations:

- Common snapshot does not exist between the source dataset and the target dataset, and none of the child datasets of the target dataset have snapshots replicated from the corresponding child dataset of the source dataset.

For example, suppose the source dataset is `src-pool1/app1-ds1` and the target dataset is `targ-pool1/app1-ds1-copy`. There is no common snapshot between the following datasets:

- `src-pool1/app1-ds1` and `targ-pool1/app1-ds1-copy`

- `src-pool1/app1-ds1/fs1` and `targ-pool1/app1-ds1-copy/fs1`
- `src-pool1/app1-ds1/vol1` and `targ-pool1/app1-ds1-copy/vol1`

In such situation, if you use the replication stream package option, the replication module uses full snapshot streams of all these datasets to create a replication stream package, and receives that stream package on the target dataset. This removes any existing data on the target dataset or its child datasets, and replicates data from the source dataset hierarchy.

- Name of the latest replicated snapshot on the source and target datasets is the same as the name of the latest common snapshot on each pair of their corresponding child datasets. In such situation, the module will use incremental replication streams of all the datasets to create an incremental replication stream package.

For example, suppose the source dataset is `src-pool1/app1-ds1` and the target dataset is `targ-pool1/app1-ds1-copy`, and the latest common replicated snapshot among each pair of datasets in the source and target dataset hierarchy is `snap1`. That is, the following common snapshot exists on the source and target dataset pairs in the hierarchy, and there is no other replicated snapshot on any of those dataset pairs that is newer than `snap1`:

- `src-pool1/app1-ds1@snap1` and `targ-pool1/app1-ds1-copy@snap1`
- `src-pool1/app1-ds1/fs1@snap1` and `targ-pool1/app1-ds1-copy/fs1@snap1`
- `src-pool1/app1-ds1/vol1@snap1` and `targ-pool1/app1-ds1-copy/vol1@snap1`

In such situation, the replication module performs incremental replication by using `snap1` as the base snapshot in the incremental replication stream package for this hierarchy.

If the latest replicated snapshots on every corresponding pair of datasets in the source and target dataset hierarchy do not have the same snapshot name, then you cannot use replication stream package to replicate the hierarchy together. Instead, you must use the recursive stream package.

Recursive Stream Package

Recursive stream package is created using `-r` option with the `zfs send` command. You can choose this type of replication by using the `RECURSIVE` parameter in the replication parameters configuration file. For more information about the parameters available in the configuration file, see [“Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File” on page 26](#).

You can use recursive stream package in one of the following situations:

- Common snapshot does not exist between the source dataset and the target dataset, and none of the child datasets of the target dataset have snapshots replicated from the corresponding child dataset of the source dataset.

- Name of the latest replicated snapshot on the source and target datasets is the same as the name of the latest common snapshot on each pair of their corresponding child datasets. In such situation, the module will use incremental replication streams of all the datasets to create an incremental recursive stream package.
- Name of the latest replicated snapshots on every corresponding pair of datasets in the source and target dataset hierarchy is not the same.

For example, suppose the latest replicated snapshot between `src-pool1/app1-ds1` and `targ-pool1/app1-ds1-copy` is `snap1`. However, the latest replicated snapshot between `src-pool1/app1-ds1/fs1` and `targ-pool1/app1-ds1-copy/fs1` is `snap2`. In this situation, the module first individually replicates each dataset in the hierarchy on to the target so that the target dataset's hierarchy has a common latest replicated snapshot. Then, the replication module uses recursive stream package for subsequent replication attempts.

Note - If you want to use replication stream package, but you have different latest replicated snapshots on the target dataset hierarchy, then set up replication initially using recursive stream package. This will enable the replication to synchronize the target dataset hierarchy with the source dataset hierarchy, so that all datasets in the target dataset's hierarchy have the latest common replicated snapshot name. Then, modify the replication parameters to use replication stream package.

Incremental Replication and Intermediary Snapshots

In each replication attempt, the replication module will take a new snapshot on the source dataset. It is a recursive snapshot if the replication uses replication stream package or recursive stream package. If there is a base snapshot among the source and target datasets that can be used as the base for incremental replication, then the replication module will attempt incremental replication with the stream containing incremental changes between the base snapshot and the new snapshot. Incremental replication is performed by using the `-i` option with the `zfs send` command.

The source dataset may have snapshots that are more recent than the base snapshot but are older than the new snapshot created by the replication module. If you want such intermediary snapshots to be replicated, then set the `INTERMEDIARY_SNAPSHOTS` parameter to `TRUE` in the replication parameters configuration file and setup or modify the replication component. Then the replication module will use the `-I` option with the `zfs send` command to replicate such intermediary snapshots. Note that this option can be used when replicating a dataset individually or with replication stream package, but not with recursive stream package, as the `zfs send` command does not support `-r` and `-I` options to be used together.

Sending ZFS Properties

This replication feature supports the use of `-p` option with the `zfs send` command to send ZFS properties in a snapshot stream for replication. To use this feature, set the `SEND_PROPERTIES` parameter to `TRUE` in the replication parameters configuration file. For more information about the `SEND_PROPERTIES` parameter, see [“Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File” on page 26](#).

Oracle Solaris ZFS Snapshot Replication Package

The Oracle Solaris ZFS snapshot replication feature in disaster recovery framework software is delivered through the package `ha-cluster/geo/replication/zfs-snapshot` in the Oracle Solaris Cluster IPS repository. Install this package before attempting to create Oracle Solaris ZFS snapshot replication protection groups.

Type the following command to install the package:

```
# /usr/bin/pkg install ha-cluster/geo/replication/zfs-snapshot
```

Oracle Solaris ZFS Snapshot Infrastructure Resource Groups

If an application resource group is a failover resource group, then the disaster recovery framework software creates an infrastructure resource group for each Application Resource Group whose ZFS datasets are added to a protection group for ZFS snapshot-based data replication. The name of the ZFS pool on the primary and secondary clusters can differ.

The name of an infrastructure resource group has the following format:

```
pg-AppRGName-infr-rg
```

For example, a protection group named `pg1` that uses the application resource group `app-rg1` has an infrastructure resource group named `pg1-app-rg1-infr-rg`.

If an application resource group is a scalable resource group, then the disaster recovery framework software creates an infrastructure resource group for each ZFS pool whose ZFS datasets are added to a protection group for ZFS snapshot-based data replication. The name of the ZFS pool on the primary and secondary clusters can differ.

The name of an infrastructure resource group has the following format:

pg-zpool-infr-rg

For example, a protection group named `pg1` that uses the `srcpool1` ZFS pool has an infrastructure resource group named `pg1-srcpool1-infr-rg`.

Each infrastructure resource group contains the following two resources:

- A logical hostname resource for the local logical host that is used for replication of ZFS datasets of the ZFS pool, if the local partner is a global zone.

If an application resource group is a failover resource group, then one logical hostname resource is configured per infrastructure resource group. The name of the logical hostname resource has the following format:

pg-AppRGName-lh-rs

If an application resource group is a scalable resource group, then each zpool in each cluster that is under replication using Oracle Solaris ZFS snapshot requires a logical hostname resource. the logical hostname resource has the following format:

pg-zpool-lh-rs

If local partner is a non-global zone, the logical hostname resource required for replication must reside in another resource group configured in the global zone of the local partner zone cluster instead of the infrastructure resource group. Such a resource group and resource must be created by the user. In this case, the logical hostname resource does not follow any naming format.

- An `HAStoragePlus` resource for managing the ZFS pool within the infrastructure resource group. The name of this resource has the following format:

pg-zpool-stor-rs

An `HAStoragePlus` resource is configured in infrastructure resource group for each of the zpools managed by the application resource group. If multiple zpools are managed by a single application resource group, which is a failover resource group, then all such `HAStoragePlus` resource are combined in a single infrastructure Resource group.

For examples displaying resource groups and resources, see [Appendix B, “Use Cases for Oracle Solaris ZFS Snapshot Replication”](#).

Note - The application resource group would already have a `HASStoragePlus` resource to manage the zpool used by the application. When creating replication setup, the disaster recovery framework software creates another `HASStoragePlus` resource for the same zpool. The disaster recovery framework software utilizes two properties of the `SUNW.HASStoragePlus` resource type, `ZpoolsExportOnStop` and `ZpoolsImportOnly`, to create such replication setup consisting of two `HASStoragePlus` resources for the same zpool. The disaster recovery framework software also sets a strong positive affinity with failover delegation from the application resource group to the infrastructure resource group and an offline-restart resource dependency from the application's `HASStoragePlus` resource managing the zpool to the replication infrastructure `HASStoragePlus` resource for the same zpool. The configuration of multiple `HASStoragePlus` resources for the same zpool and use of such properties of the `SUNW.HASStoragePlus` resource type is not supported outside of the disaster recovery framework ZFS snapshot replication setup. When such replication setup is created by the Geo software, a device group is also created for the same zpool. The zpool device group can be seen in the global zone of the cluster in the output of commands that display information about device groups, such as `cldevicegroup list` or `cldevicegroup status`. Note that such setup is solely for use by the disaster recovery framework ZFS snapshot replication software. Do not disable nor switchover the zpool device group directly.

Note - The application resource group would already have a `HASStoragePlus` resource to manage the zpool used by the application. While creating the replication setup, the `HASStoragePlus` resource created by the disaster recovery framework software uses only the `ZpoolImportOnly` property. Application could be configured as failover or scalable over zpools for globally mounted ZFS filesystems. For failover application, strong positive affinity with failover delegation is set from application resource group to infrastructure resource group. If the application has a `HASStoragePlus` resource configured, set an offline-restart resource dependency from the application's `HASStoragePlus` resource managing the zpool to the replication infrastructure `HASStoragePlus` resource for the same zpool. For scalable application, set weak resource dependency from the application's `HASStoragePlus` resource managing the zpool to the replication infrastructure `HASStoragePlus` resource for the same zpool. No affinities are set from application resource group to infrastructure resource group.

Oracle Solaris ZFS Snapshot Replication Agent Resource Groups

When a replication component is added to a protection group, the disaster recovery framework software creates a ZFS snapshot replication agent resource for that replication component in a ZFS snapshot replication agent resource group. The resource type of this resource is `ORCL.ZFSSnapRepAgent`, which is a new type shipped with the Oracle Solaris ZFS snapshot

replication package. This resource performs the ZFS snapshot replication associated with the replication component.

One ZFS snapshot replication agent resource group with one ZFS snapshot replication agent resource is created for each replication component.

The name of the replication agent resource group has the following format:

repcomp-snap-rg

where *repcomp* is the name of the replication component.

The name of a replication agent resource has the following format:

repcomp-snap-rs

The disaster recovery framework software sets a strong positive affinity from such a replication agent resource group to the associated Oracle Solaris ZFS snapshot infrastructure resource group, and an offline-restart resource dependency from such a replication agent resource to the infrastructure HAStoragePlus resource managing the associated ZFS pool. This is set so that the replication agent gets online on the same node as the infrastructure HAStoragePlus resource when activated and the ZFS pool is imported, and can perform ZFS commands on the associated ZFS pool as required for its replication work.

The replication agent resource group must have a strong positive affinity on the associated infrastructure resource group. This is managed by disaster recovery framework software and you do not need to manually operate on it typically.

Note - The intent of using a strong affinity without failover delegation is that a failover of the replication agent resource group must not result in the failover of the infrastructure resource group. The disaster recovery framework software sets a strong positive affinity with failover delegation from the application resource group to the infrastructure resource group in case of failover applications. If the replication agent resource group has a strong affinity with failover delegation on to the infrastructure resource group, then a failover of replication agent resource group would induce a failover of the infrastructure resource group, thereby initiating a failover of the application resource group. Thus, a failure in replication will also attempt to move the application, which may not be desirable always. Hence, the default affinity is just a strong affinity. But if you still want to set a strong positive affinity with failover delegation to ensure failure in replication attempts to failover the application and replication both to a different node, you can set such affinity manually from the replication agent resource group to the infrastructure resource group after creating the replication component. The disaster recovery framework software will not overwrite such affinity.

Oracle Solaris ZFS Snapshot Replication Status Resource Groups

When a replication component is added to a protection group, the disaster recovery framework software creates a special replication status resource for that replication component in the replication status resource group of the protection group. The resource type of this resource is `ORCL.ZFSSnapRepStatus`, which is a new type shipped with the Oracle Solaris ZFS snapshot replication package. The disaster recovery framework software uses the replication status resource groups to monitor the overall status of replication. One replication status resource group is created for a protection group, and one replication status resource in that replication status resource group is created for each replication component in the protection group. The replication status resource in the replication status resource group monitors the replication status of the replication component.

The names of the replication status resource groups and resources do not conform to any format. You must provide them in the configuration file when filling in replication parameters to set up replication.

Replicating Data in an Oracle Solaris ZFS Snapshot Protection Group (Task Maps)

This section summarizes the tasks for configuring and administering Oracle Solaris ZFS snapshot remote replication in a protection group.

TABLE 1 Configuration Tasks for Oracle Solaris ZFS snapshot Remote Replication

Task	Description
Plan the Oracle Solaris ZFS snapshot replication configuration.	See “Planning Remote Replication Using Oracle Solaris ZFS Snapshot” on page 24.
Prerequisites for configuring the Oracle Solaris ZFS snapshot replication.	See “Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot” on page 32.
Create a protection group that is configured for Oracle Solaris ZFS snapshot replication.	See “How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group” on page 34.
Add a replication component that is controlled by the Oracle Solaris ZFS snapshot software.	See “How to Add a Replication Component to an Oracle Solaris ZFS snapshot Protection Group” on page 44.
Add application resource groups to the protection group.	See “How to Add an Application Resource Group to an Oracle Solaris ZFS Snapshot Protection Group” on page 50.
Replicate the protection group configuration to a secondary cluster.	See “How to Replicate the Oracle Solaris ZFS Snapshot Protection Group Configuration to a Partner Cluster” on page 56.

Task	Description
Activate the protection group.	See “How to Activate a Protection Group” in <i>Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4</i> .
Verify the protection group configuration.	See Chapter 3, “Migrating Services That Use Oracle Solaris ZFS Snapshot Remote Replication”.
Check the runtime status of replication.	See “Checking the Runtime Status of Oracle Solaris ZFS Snapshot Remote Replication” on page 72.

TABLE 2 Administration Tasks for Oracle Solaris ZFS Snapshot Remote Replication

Task	Description
Detect failure.	See “Detecting Cluster Failure” in <i>Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4</i> .
Migrate services by using a switchover.	See “Migrating Replication Services by Switching Over Protection Groups” in <i>Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4</i> .
Migrate services by using a takeover.	See “Forcing a Takeover of a Protection Group” in <i>Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4</i> .
Recover data after forcing a takeover.	See “Recovering Services to a Cluster on a System That Uses Oracle Solaris ZFS Snapshot Replication” on page 80.

Planning Remote Replication Using Oracle Solaris ZFS Snapshot

To configure replication using Oracle Solaris ZFS snapshot, you must use the configuration file and the setup script delivered in the package `ha-cluster/geo/replication/zfs-snapshot` for setting up protection groups. The setup script uses the parameters in the configuration file to configure the Oracle Solaris ZFS snapshot replication. The setup script performs the following operations to set up the replication:

1. Creates the protection group.
2. Creates the replication component.
3. Adds the application resource groups to the protection group.

You must use the configuration file and the setup script to do the following:

- Add more snapshot replication components to an existing ZFS snapshot replication protection group.
- Create more such protection groups.

- Modify the properties of an existing ZFS snapshot replication component.

This section contains the following information:

- [“Guidelines for Remote Replication Using Oracle Solaris ZFS Snapshot” on page 25](#)
- [“Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File” on page 26](#)
- [“Disaster Recovery Framework Properties for Oracle Solaris ZFS Snapshot Replication” on page 29](#)
- [“Remote Replication Layer Process for Validating the Application Resource Groups and Remote Replication Entities” on page 30](#)

Guidelines for Remote Replication Using Oracle Solaris ZFS Snapshot

Note the following guidelines and restrictions when planning your Oracle Solaris ZFS snapshot remote replication configuration:

- **Supported versions of Oracle Solaris ZFS snapshot software** – Because Oracle Solaris ZFS snapshots is an inherent part of Oracle Solaris, disaster recovery framework ZFS snapshot replication feature has no extra restrictions related to Oracle Solaris versions other than what is mandated by the corresponding release of Oracle Solaris Cluster software.

Note - When upgrading ZFS version of a replicated pool or filesystem, you must ensure that the upgrade does not create an incompatibility of ZFS versions between the source and the replicated target that would fail the snapshot replication using the `zfs send` or `zfs receive` command. When upgrading to a ZFS version that introduces an incompatibility, upgrade both the source and the target pool or file system on the partner clusters.

- **Restriction for zone clusters as partner members** –
 - Zone clusters are supported as members of a disaster recovery framework partnership having Oracle Solaris ZFS snapshot protection groups only for failover zpools. However, you must perform certain configuration actions in the global zone and this information is covered in the corresponding procedures throughout this guide.
 - ZFS-based cluster filesystems in zone clusters require the `zfs` snapshot protection group to be configured on the global clusters. The application that uses the ZFS-based cluster filesystem in the zone cluster is in a different protection group in the zone clusters. See [Appendix C, “Configuring Oracle Solaris ZFS Snapshot Replication for ZFS-based Cluster Filesystems in Zone Clusters”](#).

- **Restriction for synchronous replication** – Replication using Oracle Solaris ZFS snapshot is asynchronous. Therefore, this data replication feature in Oracle Solaris disaster recovery framework does not support synchronous replication, which does not consider data to be committed to stable storage until it is committed to stable storage on both the primary and secondary storage systems.
- **Quorum devices** – Do not configure a replicated volume as a quorum device. Locate any quorum devices on a shared unreplicated volume or use a quorum server.
- **Replication user** – Configure a root or non-root user to perform replication using Oracle Solaris ZFS snapshot.
- **SSH setup** – Set up SSH public/private keys and passphrase for the replication user on a cluster. Ensure that the replication user on a cluster has the same SSH passphrase protecting the user's private key on all nodes of the same cluster. For more information, see [“Use Case: Setting Up Replication User and SSH” on page 100](#).

For guidelines and requirements related to Oracle Solaris ZFS snapshot, refer to the Oracle Solaris ZFS documents at [Managing ZFS File Systems in Oracle Solaris 11.4](#).

Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File

Oracle Solaris ZFS snapshot remote replication with disaster recovery framework is developed with the script-based plug-in module of disaster recovery framework. Your ZFS snapshots replication configuration must comply with all rules of the script-based plug-in. For each protection group, you must provide a script-based plug-in configuration file on each node. For more information, see [“How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group” on page 34](#).

Additionally, the replication configuration file `zfs_snap_geo_config` is shipped with disaster recovery framework ZFS snapshot replication software. Creation of the replication protection group for disaster recovery framework is an automated process that takes the ZFS snapshots replication configuration file as input and performs the necessary actions. This file provides the parameters that are used to create a protection group, replication component, and replication options. The `zfs_snap_geo_register` setup script uses the configuration file to configure the replication. The essential content of this file consists of the following key=value pairs:

PS

Name of the partnership.

PG

Name of the protection group.

REPCOMP

Name of the replication component. This name is used to create a resource group named *REPCOMP-snap-rg* and a resource named *REPCOMP-snap-rs*. An entry for this replication component must also be found in the file pointed to by *CONFIGFILE*.

REPRS

Name of the replication status resource that monitors snapshots replication.

REPRG

Name of the replication status resource group to contain the replication resource.

DESC

Description for the protection group.

APPRG

Name of the application resource group.

CONFIGFILE

Path to the script-based plug-in configuration file.

LOCAL_REP_USER

Name of the replication user on the local cluster.

REMOTE_REP_USER

Name of the replication user on the remote cluster.

LOCAL_PRIV_KEY_FILE

Path to the file that holds the SSH private key of the local replication user on the local cluster nodes. Leave this parameter empty to use the default path used by *ssh-add*.

REMOTE_PRIV_KEY_FILE

Path to the file that holds the SSH private key of the remote replication user on the remote cluster nodes. Leave this parameter empty to use the default path used by *ssh-add*.

LOCAL_ZPOOL_RS

Name of the HAStoragePlus resource in the application managing the ZFS pool on the local cluster.

REMOTE_ZPOOL_RS

Name of the HAStoragePlus resource in the application managing the ZFS pool on the remote cluster.

LOCAL_LH

Logical hostname on local cluster to be used as hostname for replication.

REMOTE_LH

Logical hostname on remote cluster to be used as hostname for replication.

Note - If an application resource group is failover and manages multiple zpools, one `local_lh` and one `remote_lh` is required for one such application resource group. If an application resource group is scalable, one `local_lh` and one `remote_lh` should be specified for each zpools managed by the application resource group.

LOCAL_DATASET

ZFS dataset on the local cluster that is put under replication

REMOTE_DATASET

ZFS dataset on the remote cluster that is put under replication. If you do not provide a value for the ZFS dataset on the remote cluster, it uses the `LOCAL_DATASET` value as the default value.

REPLICATION_INTERVAL

Interval between ZFS snapshot replication cycles in seconds. The default value is 120. Minimum value is 0.

NUM_OF_SNAPSHOTS_TO_STORE

Number of successfully replicated snapshots to keep on the local and remote sides. The default value is 2. Minimum value is 1.

Note - Use a low value for `NUM_OF_SNAPSHOTS_TO_STORE`. Storing more replication snapshots increases the duration of the replication cycle and also space consumption in the ZFS pools on the partner clusters.

REPLICATION_STREAM_PACKAGE

Specifies whether the replication should use replication stream package. The default value is `FALSE`.

SEND_PROPERTIES

Specifies whether ZFS dataset properties must be part of the snapshot streams. The default value is `TRUE`.

INTERMEDIARY_SNAPSHOTS

Specifies whether to send intermediary snapshots during incremental replication. The default value is FALSE.

RECURSIVE

Specifies whether to replicate the entire hierarchy of the local dataset. The default value is TRUE.

MODIFY_PASSPHRASE

Specifies whether to modify the replication user SSH passphrases, during an attempt to modify an existing replication component. The default value is FALSE.

Note - Only one parameter out of RECURSIVE and INTERMEDIARY_SNAPSHOTS can be TRUE at the same time. Only one parameter out of RECURSIVE and REPLICATION_STREAM_PACKAGE can be TRUE at the same time.

For more information, see [Chapter 13, “Script-Based Plug-Ins” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Disaster Recovery Framework Properties for Oracle Solaris ZFS Snapshot Replication

This section describes the properties that can be modified for Oracle Solaris ZFS snapshot remote replication.

The following table lists the general properties.

Property Type	Properties
General protection group properties	<ul style="list-style-type: none"> ■ RoleChange_ActionCmd ■ Timeout ■ External_Dependency_Allowed ■ RoleChange_ActionArgs <p>See “Protection Group Properties” in Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4.</p>

You can run the `zfs_snap_geo_register` script with the changed configuration file to modify properties that are specified in the configuration file. For more information, see [“How to Modify an Oracle Solaris ZFS Snapshot Replication Component” on page 59](#). To change general protection group properties, use the `geopg set-prop` command.

Remote Replication Layer Process for Validating the Application Resource Groups and Remote Replication Entities

During protection group validation, the Oracle Solaris ZFS snapshot remote replication layer validates the replication entities such as resource groups and resources relevant to the replication setup. The disaster recovery framework software creates the replication resource group and its resources, if they do not already exist. If a resource group or resource of the same name already exists, the disaster recovery framework software might modify its properties. The software cannot create a new resource group or a resource of the same name if one already exists. The software also validates the affinity and dependency from the application to the replication infrastructure as required to include it in the replication setup. For more information about the resource groups, resources, and their affinities and dependencies, see [“Overview of Data Replication Using Oracle Solaris ZFS Snapshot” on page 11](#) and [Appendix B, “Use Cases for Oracle Solaris ZFS Snapshot Replication”](#).

Ensure that the `Auto_start_on_new_cluster` property is set to `FALSE` on the application resource group.

If the partner cluster is a zone cluster, ensure that the replication logical hostname resource group and resource in the global zone has the correct affinity and dependency. For information about affinity and dependency, see [“How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group” on page 34](#). Also, ensure that the `Auto_start_on_new_cluster` property is set to `TRUE` on that logical hostname resource group.

During protection group validation, the Oracle Solaris ZFS snapshot remote replication layer validates the replication configuration. It verifies and modifies the replication configuration to ensure that the following configuration conditions are met:

- Application resource groups in the protection group have the `Auto_start_on_new_cluster` property set to `False`.
- Appropriate resource group affinities, resource properties, and dependencies exist on the various resource groups and resources that are part of the protection group.

When you bring a protection group online, the application resources groups participating in that protection group are brought online only on the protection group's primary cluster. Setting the `Auto_start_on_new_cluster` property to `False` prevents the Oracle Solaris Cluster Resource Group Manager (RGM) from automatically starting the application resource groups. In this case, the startup of resource groups is reserved for the disaster recovery framework software.

Application resource groups must be online only on the primary cluster when the protection group is activated.



Caution - Do not change, remove, or take offline infrastructure or replication resource groups and resources. Use only disaster recovery framework commands to administer infrastructure and replication resource groups and resources that are internal entities managed by disaster recovery framework software. Altering the configuration or state of these entities directly with Oracle Solaris Cluster commands might result in unrecoverable failure.

At creation time, when the validation is complete, the disaster recovery framework software adds the application resource group to the protection group if not added already.

Creating and Validating an Oracle Solaris ZFS Snapshot Protection Group

This section contains the following topics:

- [“Strategies for Creating Oracle Solaris ZFS Snapshot Protection Groups” on page 31](#)
- [“Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot” on page 32](#)
- [“How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group” on page 34](#)
- [“Debugging an Oracle Solaris ZFS Snapshot Protection Group” on page 40](#)

Strategies for Creating Oracle Solaris ZFS Snapshot Protection Groups

You can perform the steps described in the following task map to create Oracle Solaris ZFS snapshot protection groups:

TABLE 3 Tasks for Creating a Protection Group

Task	Description
Create the protection group from a cluster node and add replication component and application resource group to it.	See “How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group” on page 34 .
Add more replication components to a protection group.	See “How to Add a Replication Component to an Oracle Solaris ZFS snapshot Protection Group” on page 44 .
Add more application resource groups to a protection group.	See “How to Add an Application Resource Group to an Oracle Solaris ZFS Snapshot Protection Group” on page 50 .

Task	Description
From the secondary cluster, retrieve the protection group configuration.	See “How to Replicate the Oracle Solaris ZFS Snapshot Protection Group Configuration to a Partner Cluster” on page 56.
Start the protection group.	See “How to Activate a Protection Group” in <i>Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4</i> .

Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot

Perform the following actions before you run the setup script to configure the Oracle Solaris ZFS snapshot replication:

- Since the application uses ZFS as data store, the application resource group would already have a HAStoragePlus resource managing the ZFS pool used by the application. If the application resource group does not have a HAStoragePlus resource, create a HAStoragePlus resource to manage the ZFS pool that the application uses on both primary and secondary cluster. Provide the name of that HAStoragePlus resource in the configuration file when configuring replication of datasets of the ZFS pool. Ensure to upgrade the HAStoragePlus resource to at least version 11 of the HAStoragePlus resource type. For information about creating an HAStoragePlus, see [Planning and Administering Data Services for Oracle Solaris Cluster 4.4](#).
- To use zpools for globally mounted ZFS filesystems, you must configure a device group for the zpool. The name of such a zpool device group must be the same as the zpool. The application resource group might have a HAStoragePlus resource configured in it. If the application resource group does not have a HAStoragePlus resource, such entries will be empty in the configuration file in case of zpools for globally mounted ZFS filesystems.
- Decide which hostnames to use as logical hostname for replication infrastructure on each partner cluster.
- In the global zone on all nodes in the application's node-list on both clusters, configure the replication user with ZFS permissions to perform the required ZFS operations on the ZFS datasets that are added to the protection group. Ensure that the replication user is configured in the global zone on all nodes in the application's node-list on partner clusters. The ZFS permissions are needed in the global zone on all nodes in the application's node-list on partner clusters.

The user must have the following Local+Descendent ZFS permissions on the ZFS dataset in the cluster where the user is set up : create, destroy, hold, mount, receive, release, rollback, send, and snapshot. For an example illustrating how to set the ZFS permission,

see step 3 in [“Use Case: Configuring Oracle Solaris ZFS Snapshot Replication When Both Partner Clusters Are Global Zones”](#) on page 108

For more information about ZFS dataset permissions, see [Oracle Solaris ZFS Delegated Administration](#).

The source and target datasets must already exist in the source and target zpools on the primary and secondary clusters.

- Perform the SSH configuration to enable the replication user to communicate between global zones of partner clusters. Perform the following actions to set up SSH:
 - Set up SSH keys for the replication user in global zone of each partner cluster, and copy the public keys on to the corresponding remote global zone of partner cluster.
 - Specify the SSH passphrase as input to the replication setup. Note the following when providing the passphrase:
 - If both partners are global zones, the setup or register script will prompt for SSH passphrases when executed to create a snapshot replication component. If the `MODIFY_PASSPHRASE` parameter is set to `True` in the configuration file, the setup or register script will prompt for such passphrases when executed to modify an existing snapshot replication component. Provide the passphrase for local cluster and remote cluster replication user at the prompts.
 - If at least one partner is a zone cluster, you must configure an Oracle Solaris Cluster private string object in the global zone of each partner to store the passphrase for the private key of the replication user of the partner. Note that the Oracle Solaris Cluster private string must be created even for a global zone partner if its partner is a zone cluster.

The name of the Oracle Solaris Cluster private string must have the following format:

```
local-partner-zonename:replication-component:local_service_passphrase
```

For information about generating a public/private key pair for use with secure shell, see [How to Generate a Public/Private Key Pair for Use With Secure Shell](#). For an example illustrating how to set up SSH, see [“Use Case: Setting Up Replication User and SSH”](#) on page 100.

- You must ensure that the application resource groups that are added to the protection group are in the unmanaged state on the primary and secondary clusters.

When you configure remote replication using ZFS snapshot for an application, the setup software sets resource group affinities on the application resource group and also sets resource properties on the `HASStoragePlus` resource that manages the zpool in the application. The setup script also adds the application to the protection group. To perform all the required configuration set up, it is required that the application is in the unmanaged state on the primary and secondary clusters when creating the replication component.

- For a zone cluster, before upgrading Oracle Solaris Cluster from 4.3 to 4.4, the resource dependencies of logical hostname resource on the HAStoragePlus resource of infrastructure resource group and the resource group affinities of logical hostname resource group on infrastructure resource group needs to be removed.

For example, suppose the local partner *paris* is a zone cluster and the logical hostname resource group in global zone is *paris-lh-rg*, and the resource is *paris-lh-rs*. Type the following commands in the global zone of one node of *paris* to only remove the affinity and dependency:

```
$ clrg set -p RG_affinities-=+paris:pg1-srcpool1-infr-rg paris-lh-rg
$ clrs set -p Resource_dependencies_offline_restart-=paris:pg1-srcpool1-stor-rs
paris-lh-rs
```

▼ How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group

Before You Begin Ensure that the following conditions are met:

- The disaster recovery framework software is installed on the primary and secondary clusters.
- You have reviewed the information in [“Planning Remote Replication Using Oracle Solaris ZFS Snapshot” on page 24](#).
- You have performed the prerequisites mentioned in [“Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot” on page 32](#).
- The local cluster is a member of a partnership.
- The protection group you are creating does not already exist on either partner cluster.
- The application resource group containing the HAStoragePlus resource managing the zpool is in the unmanaged state on the primary and secondary clusters.

1. **Assume the root role or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see [“Securing Disaster Recovery Framework Software” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Note - If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwX:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management RBAC rights profile and Oracle Solaris ZFS snapshot software.

2. On all nodes of both clusters, create or update the script-based plugin configuration file for a protection group. This file contains the details of which nodes pertain to a replication component in the protection group.

Update the file so that it contains one line that contains the rule information for the replication component.

```
replication-component|any|nodelist
```

```
replication-component
```

Name of the replication component provided in the replication configuration file.

```
nodelist
```

The name of one or more cluster nodes where the plug-in is to validate the configuration.

Ensure to add `any` as the evaluation rule.

For example, suppose that you want to create and use a file `/var/tmp/geo/zfs_snapshot/sbp_conf`. Suppose that the nodes of cluster `paris` are `paris-node-1` and `paris-node-2`. On each node of the cluster `paris`, type the following commands:

```
paris-node-N# mkdir -p /var/tmp/geo/zfs_snapshot
paris-node-N# echo "repcom1|any|paris-node-1,paris-node-2" > /var/tmp/geo/zfs_snapshot/
sbp_conf
```

Suppose that the nodes of the cluster `newyork` are `newyork-node-1` and `newyork-node-2`. On each node of cluster `newyork`, type the following commands:

```
newyork-node-N# mkdir -p /var/tmp/geo/zfs_snapshot
newyork-node-N# echo "repcom1|any|newyork-node-1,newyork-node-2" > /var/tmp/geo/
zfs_snapshot/sbp_conf
```

For more information about configuration files, see [“configuration_file Property” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

3. **Ensure that the `Auto_start_on_new_cluster` property of the application resource group is set to `False`.**

```
# clresourcegroup show -p Auto_start_on_new_cluster app-group
```

If necessary, change the property value to `False`.

```
# clresourcegroup set -p Auto_start_on_new_cluster=False app-group
```

4. **If either partner is a zone cluster, then configure an Oracle Solaris Cluster private string in the global zone on each partner to store the SSH passphrase of the replication user on that partner.**

The name of the private string must have the following format :

```
local-partner-zonename:replication-component:local_service_passphrase
```

For example:

- Partnership between a global zone and a zone cluster – Suppose the name of the zone cluster is `zc1`. The name of the replication component is `repcom1`. The replication user for the global zone partner is `zfsuser1`. The replication user for the zone cluster partner is `zfsuser2`.

In one node of the global zone partner, type the following command to create a private string to store the SSH passphrase of `zfsuser1`:

```
# clps create -b global:repcom1:local_service_passphrase \  
global:repcom1:local_service_passphrase  
<Enter SSH passphrase for zfsuser1 at prompt>
```

In the global zone of one node of the zone cluster partner `zc1`, type the following command to create a private string to store the SSH passphrase of `zfsuser2` :

```
# clps create -b zc1:repcom1:local_service_passphrase \  
zc1:repcom1:local_service_passphrase  
<Enter SSH passphrase for zfsuser2 at prompt>
```

- Partnership between two zone clusters – Suppose the partnership is between zone clusters `zc1` and `zc2` and the replication component is `repcom1`. Suppose that the replication user for `zc1` is `zfsuser1` and that for `zc2` is `zfsuser2`.

In the global zone of one node of the zone cluster partner `zc1`, type the following command to create a private string to store the SSH passphrase of `zfsuser1`:

```
# clps create -b zc1:repcom1:local_service_passphrase \  
zc1:repcom1:local_service_passphrase  
<Enter SSH passphrase for zfsuser1 at prompt>
```

In the global zone of one node of the zone cluster partner zc2, type the following command to create a private string to store the SSH passphrase of zfsuser2 :

```
# clps create -b zc2:repcom1:local_service_passphrase \  
zc2:repcom1:local_service_passphrase  
<Enter SSH passphrase for zfsuser2 at prompt>
```

5. On one node of the primary cluster, copy the default replication configuration file to another location and specify the values in the file.

For example, copy the file to /var/tmp/geo/zfs_snapshot directory.

```
# cp /opt/ORCLscgprepzfssnap/etc/zfs_snap_geo_config /var/tmp/geo/zfs_snapshot
```

The following list uses sample values:

```
PS=paris-newyork  
PG=pg1  
REPCOMP=repcom1  
REPRS=repcom1-repstatus-rs  
REPRG=pg1-repstatus-rg  
DESC="Protect app1-rg1 using ZFS snapshot replication"  
APPRG=app1-rg1  
CONFIGFILE=/var/tmp/geo/zfs_snapshot/sbp_conf  
LOCAL_REP_USER=zfsuser1  
REMOTE_REP_USER=zfsuser2  
LOCAL_PRIV_KEY_FILE=  
REMOTE_PRIV_KEY_FILE=  
LOCAL_ZPOOL_RS=par-app1-hasp1  
REMOTE_ZPOOL_RS=ny-app1-hasp1  
LOCAL_LH=paris-lh  
REMOTE_LH=newyork-lh  
LOCAL_DATASET=srcpool1/app1-ds1  
REMOTE_DATASET=targpool1/app1-ds1-copy  
REPLICATION_INTERVAL=120  
NUM_OF_SNAPSHOTS_TO_STORE=2  
REPLICATION_STREAM_PACKAGE=false  
SEND_PROPERTIES=true  
INTERMEDIARY_SNAPSHOTS=false  
RECURSIVE=true  
MODIFY_PASSPHRASE=false
```

For more information about the zfs_snap_geo_config file, see [“Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File”](#) on page 26.

- 6. On the primary cluster node where the replication configuration file with replication parameter values is stored, execute the setup script `/opt/ORCLscgrepzfssnap/util/zfs_snap_geo_register`.**

For example:

```
paris-node-1# /opt/ORCLscgrepzfssnap/util/zfs_snap_geo_register -f \  
/var/tmp/geo/zfs_snapshot/zfs_snap_geo_config
```

This setup action performed by the `zfs_snap_geo_register` script creates the following components:

- Protection group `pg1`
- Replication component `repcom1`
- Infrastructure resource group `pg1-srcpool1-infr-rg`
- Replication resource group `repcom1-snap-rg` which contains the resource `repcom1-snap-rs`
- Replication status resource group `pg1-repstatus-rg` and replication status resource `repcom1-repstatus-rs`

For details about an example setup involving resource groups and resources, see [Appendix B, “Use Cases for Oracle Solaris ZFS Snapshot Replication”](#).

- 7. Replicate the protection group to the partner cluster.**

The final messages of the setup script outline the required `geopg get` command. You must log in to one node of the partner cluster and execute that exact command.

For example, where `paris-newyork` is the partnership name and `pg1` is the protection group name:

```
newyork-node-1# geopg get --partnership paris-newyork pg1
```

- 8. If any partner is a zone cluster, configure a logical hostname resource and resource group in the global zone of that zone cluster partner to host the replication hostname. Perform this configuration for each zone cluster partner in a partnership. The names of the resource and resource group are not restricted to any specific format. After configuring the logical hostname resource and resource group, perform the following actions:**

- Add a strong positive affinity from the logical hostname resource group to the `zpool`'s infrastructure resource group.

Note - Setting the strong positive resource group affinity will print a warning message, if the logical hostname resource group has the `Auto_start_on_new_cluster=TRUE` property while the zpool's infrastructure resource group has `Auto_start_on_new_cluster=FALSE`. This is allowed, since the disaster recovery framework software will bring up the zpool's infrastructure resource group when required, thereby also bringing up the logical hostname resource group due to the affinity.

As one infrastructure resource group is configured per application resource group if it is failover, one logical hostname resource group is required for each such infrastructure resource group.

If an application resource group is scalable, one logical hostname resource group is configured for each of the zpools managed by the application resource group.

- Add an offline-restart resource dependency from the logical hostname resource to the zpool's infrastructure storage `HASStoragePlus` resource.
- Ensure that `Auto_start_on_new_cluster` is `TRUE` on the logical hostname resource group. This property is `TRUE` by default. In case the property is `FALSE`, set it to `TRUE`.

It is essential to have a strong positive affinity from such a logical hostname resource group to the associated Oracle Solaris ZFS snapshot infrastructure resource group. This must be set so that the replication logical hostname is online in the global zone of the same cluster node where the associated ZFS pool is imported by the infrastructure `SUNW.HASStoragePlus` resource.

For example:

Suppose the local partner is a zone cluster `zc1` and local replication hostname is `paris-lh`. The zpool infrastructure resource group in `zc1` is `pg1-srcpool1-infr-rg`. The storage resource is `pg1-srcpool1-stor-rs`. Type the following commands in the global zone of one node of `zc1`:

```
# clrg create paris-lh-rg
# clrslh create -g paris-lh-rg -h paris-lh paris-lh-rs
# clrg manage paris-lh-rg
# clrg set -p RG_affinities=++zc1:pg1-srcpool1-infr-rg paris-lh-rg
(C538594) WARNING: resource group global:paris-lh-rg has a strong positive affinity on
resource group zc1:pg1-srcpool1-infr-rg with Auto_start_on_new_cluster=FALSE;
global:paris-lh-rg will be forced to remain offline until its strong affinities are
satisfied.
# clrs set -p Resource_dependencies_offline_restart=zc1:pg1-srcpool1-stor-rs paris-lh-
rs
# clrg show -p Auto_start_on_new_cluster paris-lh-rg
=== Resource Groups and Resources ===
Resource Group:          paris-lh-rg
Auto_start_on_new_cluster:  True
```

If the property is not True, type the following command :

```
$ clrg set -p Auto_start_on_new_cluster=True paris-lh-rg
```

9. Verify the protection group configuration.

For example, suppose repcom1-repstatus-rs is the replication status resource name:

```
paris-node-1# geoadm status
paris-node-1# clresource status repcom1-repstatus-rs
newyork-node-1# geoadm status
newyork-node-1# clresource status repcom1-repstatus-rs
```

See Also

Note - Save the `/var/tmp/geo/zfs_snapshot/zfs_snap_geo_config` file for possible future use. When you want to modify any properties for this replication component that you created, you can edit the desired parameters in this same file and re-run the `zfs_snap_geo_register` script. For more information, see [“How to Modify an Oracle Solaris ZFS Snapshot Replication Component” on page 59](#).

Troubleshooting

If you experience failures while performing this procedure, enable debugging. See [“Debugging an Oracle Solaris ZFS Snapshot Protection Group” on page 40](#).

Next Steps

For information about activating a protection group, see [“How to Activate a Protection Group” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Debugging an Oracle Solaris ZFS Snapshot Protection Group

If you encounter problems when creating a protection group and replication component, modifying a replication component, or replicating a protection group with the `geopg get` command, you can set the `SET_DEBUG` property of the `/opt/ORCLscgrepzfssnap/etc/config` file to run trace logs. These logs will display on the terminal.

If you encounter problems with the Oracle Solaris ZFS snapshot agent or the status resources after the replication configuration is setup, then you can enable debugging directly on the resources by setting the `Debug_level` property of the appropriate resource with the `clresource set` command. Debug messages will be displayed according to the level value. The `ORCL.ZFSSnapRepAgent` and `ORCL.ZFSSnapRepStatus` resource types follow this kind of debugging control.


```
# clresource set -p Debug_Level=N zfsnap-replication-resource
```

The following values are valid for the Debug_Level properties:

- | | |
|---|-----------------------------------------------------------------------------|
| 0 | No debug messages |
| 1 | Logs debug messages into syslog about the Begin and End of functions |
| 2 | Logs all debug messages to syslog and directs KSH tracing output to console |
| 3 | Logs all debug messages to the trace log file |

Additionally, each resource of type ORCL.ZFSSnapRepAgent and ORCL.ZFSSnapRepStatus logs important messages into the trace log file and resource log file.

Trace Log File

A trace log file logs all important messages showing a resource's work progress. Such trace messages are logged irrespective of the Debug_Level of the resource, thereby ensuring that critical messages about resource's execution are always logged and stored.

A resource of type ORCL.ZFSSnapRepAgent logs messages about replication activities, such as finding common base snapshot, taking snapshots, and sending and receiving snapshot streams. A resource of type ORCL.ZFSSnapRepStatus logs messages about the actions to query and report replication status.

The location of the trace log file on a cluster node is `/var/cluster/logs/DS/Resource_Type/message_log.resource_name`. For example, the location of the trace log file for the resource rs1 of ORCL.ZFSSnapRepAgent type is `/var/cluster/logs/DS/ORCL.ZFSSnapRepAgent/message_log.rs1`. The location of the trace log file for the resource rs2 of the ORCL.ZFSSnapRepStatus type is `/var/cluster/logs/DS/ORCL.ZFSSnapRepStatus/message_log.rs2`.

Note - The trace log files and resource log files are log-rotated by `logadm`.

Resource Log File

A resource creates a log file to display the status of any command that is executed by the resource. You can examine the log file to check the errors encountered by a command and

the possible cause of failure. The location of the resource log file on a cluster node is `/var/cluster/logs/DS/Resource_Type/resource_log.resource_name`. You must examine the resource log file for run-time errors, in addition to syslog and console messages. If you require more information, examine the trace log file.

For example, the location of the resource log file for the resource `rs1` of the `ORCL.ZFSSnapRepAgent` type is `/var/cluster/logs/DS/ORCL.ZFSSnapRepAgent/resource_log.rs1`. The location of the resource log file for the resource `rs2` of the `ORCL.ZFSSnapRepStatus` type is `/var/cluster/logs/DS/ORCL.ZFSSnapRepStatus/resource_log.rs2`.

Starting an Oracle Solaris ZFS Snapshot Protection Group

Ensure that the protection group and any replication components have been correctly created. The replication infrastructure involved in a protection group includes the SSH setup, ZFS dataset permissions for the replication user, and the replication logical hostname setup in the global zone for a zone cluster partner. For more information about creating and configuring the protection group, see [“How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group” on page 34](#).

You can start a protection group locally or globally, with or without data replication. For information about activating a protection group, see [“How to Activate a Protection Group” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

When you start data replication on a cluster for a Oracle Solaris ZFS snapshot protection group, the following actions occur for each replication component in the protection group:

- The infrastructure resource group and its resources for the replication component are brought online on that cluster, if they are not online already. This makes the zpool and the replication hostname available on that cluster.
- If the role of the protection group on the local cluster is primary, the replication agent resource group and resource for the replication component are brought online. If the replication agent resource group was suspended, it is resumed back.

The replication agent resource starts up the SSH agent for its use and adds the local replication user's private key to the agent using the local SSH passphrase. If the remote replication hostname is reachable and the remote zpool is available, the replication agent resource attempts its replication work as per the replication parameters. The resource checks if there is any common snapshot between the local and remote dataset. If there is a common snapshot, it attempts to perform incremental replication, otherwise it attempts full replication. For more information about the types of replication, see [“About Oracle Solaris ZFS Snapshots Replication” on page 14](#).

- The replication status resource group is brought online if it is not online already. If the replication status resource for the replication component is not online, it is brought online. If the replication status resource group was suspended, it is resumed back.

The replication status resource starts up the SSH agent for its use and adds the local replication user's private key to the agent using the local SSH passphrase. It then derives replication status based on factors like local zpool availability and the status of the replication agent resource of the primary cluster, and reports such replication status through its own status and status message.

Data synchronization starts only after the protection group is successfully activated with data replication enabled on both partner clusters.

If the resources fail to start as described above, then the activation of the protection group fails. Check the console and syslog, and the resource logs of the relevant resource for failure messages that will indicate the cause of the error. Fix any such problems and attempt to activate the protection group again. For more information about the error messages, see [“Checking the Runtime Status of Oracle Solaris ZFS Snapshot Remote Replication”](#) on page 72.

Stopping an Oracle Solaris ZFS Snapshot Protection Group

You can stop such a protection group locally or globally, and stop only data replication optionally. For information about deactivating the protection group, see [“How to Deactivate a Protection Group”](#) in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

You must stop a ZFS snapshot protection group on the primary cluster first, before stopping it on the secondary cluster. This ensures that data replication actions from the primary are stopped first before the replication infrastructure is taken offline from the secondary cluster. To do this, you can either stop the protection group with local scope on primary first and then on secondary, or you can stop the protection group with global scope from any partner cluster.

When you stop data replication on a cluster for a Oracle Solaris ZFS snapshot protection group, the following actions happen for each replication component in the protection group:

- The replication status resource group and replication resources are taken offline. Any SSH agents started by such resources are stopped too. The replication status resource group is suspended.
- If the role of the protection group on the local cluster is primary, the replication agent resource group and resource for the replication component are taken offline. SSH agent for the resource is stopped too. The replication agent resource group is suspended.
- The infrastructure resource group and its resources for the replication component are taken offline on that cluster.

Adding a Replication Component to an Oracle Solaris ZFS Snapshot Protection Group

A protection group is the container for the application resource groups and replication components, which contain data for services that are protected from disaster. A ZFS snapshot replication component in a protection group protects the data by replicating it from the primary cluster to the secondary cluster. The software also monitors the replication status.

This section provides information about adding a replication component to an Oracle Solaris ZFS snapshot protection group:

▼ How to Add a Replication Component to an Oracle Solaris ZFS snapshot Protection Group

Perform this procedure to add a replication component to an existing Oracle Solaris ZFS snapshot protection group.

Note - When the protection group is initially created, the replication component specified in the `zfs_snap_geo_config` configuration file is added to the protection group. Thus, you only need to run this procedure to add more replication components to existing Oracle Solaris ZFS snapshot protection groups.

Before You Begin Before you add a replication component to a protection group, ensure that the following conditions are met:

- The disaster recovery framework software is installed on the primary and secondary clusters.
- You have reviewed the information in [“Planning Remote Replication Using Oracle Solaris ZFS Snapshot” on page 24](#).
- You have performed the prerequisites mentioned in [“Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot” on page 32](#).
- The local cluster is a member of a partnership.
- The protection group is defined on the local cluster.
- The protection group is offline on the local cluster and the partner cluster, if the partner cluster can be reached.
- The zpool for the dataset to be put under replication is managed by a `SUNW.HASStoragePlus` resource in an application resource group. Ensure that the application resource group is in the unmanaged state on the local partner, and also on the remote partner cluster if it is reachable.

1. Assume the root role or assume a role that is assigned the Geo Management RBAC rights profile.

For more information about RBAC, see [“Securing Disaster Recovery Framework Software” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Note - If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwX:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management RBAC rights profile and Oracle Solaris ZFS snapshot software.

2. On all nodes of both clusters, update the script-based plugin configuration file for a protection group. This file is already associated with the protection group and would be shown in the output of `geopg show`. This file contains the details of which nodes pertain to a replication component in the protection group.

Update the file so that it contains one line that contains the rule information for the replication component.

```
replication-component | any | nodelist
```

```
replication-component
```

Name of the replication component provided in the replication configuration file.

```
nodelist
```

The name of one or more cluster nodes where the plug-in is to validate the configuration.

For example, suppose the configuration file is `/var/tmp/geo/zfs_snapshot/sbp_conf`. Suppose that the nodes of cluster `paris` are `paris-node-1` and `paris-node-2`. On each node of the cluster `paris`, type the following commands:

```
paris-node-N# echo "repcom1|any|paris-node-1,paris-node-2" >> /var/tmp/geo/zfs_snapshot/  
sbp_conf
```

Suppose that the nodes of the cluster `newyork` are `newyork-node-1` and `newyork-node-2`. On each node of cluster `newyork`, type the following commands:

```
newyork-node-N# echo "repcom1|any|newyork-node-1,newyork-node-2" >> /var/tmp/geo/  
zfs_snapshot/sbp_conf
```

For more information about configuration files, see “[configuration_file Property](#)” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

3. **Ensure that the `Auto_start_on_new_cluster` property of the application resource group is set to `False`.**

```
# clresourcegroup show -p Auto_start_on_new_cluster app-group
```

If necessary, change the property value to `False`.

```
# clresourcegroup set -p Auto_start_on_new_cluster=False app-group
```

4. **If either partner is a zone cluster, configure an Oracle Solaris Cluster private string in the global zone on each partner to store the SSH passphrase of the replication user on that partner.**

The name of the private string must have the following format :

```
local-partner-zonename:replication-component:local_service_passphrase
```

For example, suppose the partnership is between a global zone and a zone cluster `zc1`. The name of the replication component is `repcom1`. The replication user for the global zone partner is `zfsuser1`. The replication user for the zone cluster partner is `zfsuser2`. In one node of the global zone partner, type the following command to create a private string to store the SSH passphrase of `zfsuser1`:

```
$ clps create -b global:repcom1:local_service_passphrase global:repcom1:
local_service_passphrase
<Enter SSH passphrase for zfsuser1 at prompt>
```

In the global zone of one node of the zone cluster partner `zc1`, type the following command to create a private string to store the SSH passphrase of `zfsuser2`:

```
$ clps create -b zc1:repcom1:local_service_passphrase zc1:repcom1:
local_service_passphrase
<Enter SSH passphrase for zfsuser2 at prompt>
```

If the partnership is between two zone clusters `zc1` and `zc2` and the replication component is `repcom1`. Suppose that the replication user for `zc1` is `zfsuser1` and that for `zc2` is `zfsuser2`. In the global zone of one node of the zone cluster partner `zc1`, type the following command to create a private string to store the SSH passphrase of `zfsuser1`:

```
$ clps create -b zc1:repcom1:local_service_passphrase zc1:repcom1:
local_service_passphrase
<Enter SSH passphrase for zfsuser1 at prompt>
```

In the global zone of one node of the zone cluster partner zc2, type the following command to create a private string to store the SSH passphrase of zfsuser2:

```
$ clps create -b zc2:repcom1:local_service_passphrase zc2:repcom1:
local_service_passphrase
<Enter SSH passphrase for zfsuser2 at prompt>
```

5. **Add a replication component to the protection group. On one node of the local cluster, copy the default replication configuration file to another location and specify the values for the replication component in the file. Then, use the zfs_snap_geo_register script with the new configuration file.**

For example, copy the file to /var/tmp/geo/zfs_snapshot directory. Use the zfs_snap_geo_register script with the new configuration file.

```
paris-node-1# cp /opt/ORCLscgprepzfssnap/etc/zfs_snap_geo_config /var/tmp/geo/
zfs_snapshot/repcom1_config
```

After you fill in the values in the configuration file, execute the setup script.

```
paris-node-1# /opt/ORCLscgprepzfssnap/util/zfs_snap_geo_register -f /var/tmp/geo/
zfs_snapshot/repcom1_config
```

This command adds a replication component to a protection group on the local cluster. If the partner cluster contains a protection group with the same name, the command also propagates the new configuration to the partner cluster.

Note - The add operation for the replication component is performed during the scripted registration. For details about scripted registration, see [“How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group”](#) on page 34.

6. **If any partner is a zone cluster, configure a logical hostname resource and resource group in the global zone of that zone cluster partner to host the replication hostname. Perform this configuration for each zone cluster partner in a partnership. The names of the resource and resource group are not restricted to any specific format.**

Note - If another replication component that uses the same zpool exists in the same protection group, then you must have configured such a logical hostname resource group for that replication component, in which case you do not need to configure another one. The existing replication component and the new replication component being configured will have the same common infrastructure resource group managing the common zpool, and the same logical hostname resource group in the global zone will suffice to co-locate the replication logical hostname with the zpool.

As one infrastructure resource group is configured per application resource group if it is failover, one logical hostname resource group is required for each such infrastructure resource group.

If an application resource group is scalable, one logical hostname resource group is configured for each of the zpools managed by the application resource group.

After configuring the logical hostname resource and resource group, perform the following actions:

- Add a strong positive affinity from the logical hostname resource group to the zpool's infrastructure resource group.

Note - Setting the strong positive resource group affinity will print a warning message, if the logical hostname resource group has the `Auto_start_on_new_cluster=TRUE` property while the zpool's infrastructure resource group has `Auto_start_on_new_cluster=FALSE`. This is allowed, since the disaster recovery framework software will bring up the zpool's infrastructure resource group when required, thereby also bringing up the logical hostname resource group due to the affinity.

- Add a offline-restart resource dependency from the logical hostname resource to the zpool's infrastructure storage `HASStoragePlus` resource.
- Ensure that `Auto_start_on_new_cluster` is `TRUE` on the logical hostname resource group. This property is `TRUE` by default. In case the property is `FALSE`, set it to `TRUE`.

It is essential to have a strong positive affinity from such a logical hostname resource group to the associated Oracle Solaris ZFS snapshot infrastructure resource group. This must be set so that the replication logical hostname is online in the global zone of the same cluster node where the associated ZFS pool is imported by the infrastructure `SUNW.HASStoragePlus` resource.

For example, suppose the local partner is a zone cluster `zc1` and local replication hostname is `paris-1h`. The zpool infrastructure resource group in `zc1` is `pg1-srcpool1-infr-rg`. The storage resource is `pg1-srcpool1-stor-rs`. Type the following commands in the global zone of one node of `zc1`:


```
# clrg create paris-lh-rg
# clrslh create -g paris-lh-rg -h paris-lh paris-lh-rs
# clrg manage paris-lh-rg
# clrg set -p RG_affinities=++zcl:pg1-srcpool1-infr-rg paris-lh-rg
(C538594) WARNING: resource group global:paris-lh-rg has a strong positive affinity on
resource group zcl:pg1-srcpool1-infr-rg with Auto_start_on_new_cluster=FALSE;
global:paris-lh-rg will be forced to remain offline until its strong affinities are
satisfied.
# clrs set -p Resource_dependencies_offline_restart=zcl:pg1-srcpool1-stor-rs paris-lh-
rs
# clrg show -p Auto_start_on_new_cluster paris-lh-rg
=== Resource Groups and Resources ===
Resource Group:          paris-lh-rg
Auto_start_on_new_cluster:  True
*****
```

If the property is not True, type the following command :

```
# clrg set -p Auto_start_on_new_cluster=True paris-lh-rg
```

7. Verify the protection group configuration.

For example, suppose repcom1- repstatus- rs is the replication status resource name:

```
paris-node-1# geoadm status
paris-node-1# clresource status repcom1-repstatus-rs
newyork-node-1# geoadm status
newyork-node-1# clresource status repcom1-repstatus-rs
```

See Also

Note - Save the `/var/tmp/geo/zfs_snapshot/repcom1_config` file for possible future use. When you want to modify any properties for this replication component that you created, you can edit the desired parameters in this same file and re-run the `zfs_snap_geo_register` script. For more information, see [“How to Modify an Oracle Solaris ZFS Snapshot Replication Component” on page 59](#).

Troubleshooting If you have difficulties adding the replication component to the protection group, see [“Debugging an Oracle Solaris ZFS Snapshot Protection Group” on page 40](#).

Adding an Application Resource Group to an Oracle Solaris ZFS snapshot Protection Group

To make an application highly available, the application must be managed as a resource in an application resource group.

Use the `zfs_snap_geo_register` script to create protection group, add replication components, and application resource groups to the protection group in one complete setup procedure. You must ensure that the parameters are filled in the configuration file before you run the register script. However, you can use the procedure in this section to add any application resource groups whose replication setup already exists in the protection group.

All the entities you configure for the application resource group on the primary cluster, such as application data resources, application configuration files, and the resource groups, must be replicated manually on the secondary cluster. The resource group names must be identical on both clusters.

▼ How to Add an Application Resource Group to an Oracle Solaris ZFS Snapshot Protection Group

Perform this procedure to add an existing resource group to the list of application resource groups in a protection group. If you specify any existing application resource groups in the `zfs_snap_geo_config` configuration file and execute the `zfs_snap_geo_register` script, such application resource groups are automatically added to the protection group and the replication configuration for such resource groups is also created in the protection group. This setup procedure is completed automatically and you are not required to add the application resource groups again to the protection group.

If the ZFS dataset corresponding to the application resource group that you want to add to the protection group has not been added to the protection group, use the `zfs_snap_geo_register` script.

However, if the ZFS dataset of the application resource group is already in the protection group, and you only want to add the application resource group to the protection group, then perform this procedure.

Note - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name, click the protection group name, and in the Resource Groups section click Add Resource Groups. For more information about Oracle Solaris Cluster Manager, see [Chapter 12, “Using the Oracle Solaris Cluster Manager Browser Interface” in *Administering an Oracle Solaris Cluster 4.4 Configuration*](#).

The protection group can be activated or deactivated and the resource group can be either online or unmanaged.

- If the resource group is unmanaged and the protection group is activated after the configuration of the protection group has changed, the local state of the protection group becomes Error.
- If the resource group to add is online and the protection group is deactivated, the request is rejected.

Before You Begin Ensure that the following conditions are met:

- The protection group is defined.
- The resource group to be added already exists on both clusters and is in an appropriate state.

1. Assume the root role or assume a role that is assigned the Geo Management RBAC rights profile.

For more information about RBAC, see [“Securing Disaster Recovery Framework Software” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Note - If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management RBAC rights profile and Oracle Solaris ZFS snapshot software.

2. Ensure that the `Auto_start_on_new_cluster` property of the resource group is set to `False`.

```
# clresourcegroup show -p Auto_start_on_new_cluster app-group
```

If necessary, change the property value to `False`.

```
# clresourcegroup set -p Auto_start_on_new_cluster=False app-group
```

- 3. If the application resource group must have dependencies on resource groups and resources that are not managed by this protection group, ensure that the `external_dependency_allowed` property of the protection group is set to `TRUE`.**

```
# geopg show protection-group | grep -i external_dependency_allowed
```

If necessary, change the property value to `True`.

```
# geopg set-prop -p external_dependency_allowed=TRUE protection-group
```

- 4. (Optional) If the protection group is deactivated, bring the application resource group to unmanaged state.**

The application resource group must be in unmanaged state to be added to a deactivated protection group.

```
# clresourcegroup offline resource-group
# clresource disable -g resource-group +
# clresourcegroup unmanage resource-group
```

- 5. Add the application resource group to the protection group.**

```
# geopg add-resource-group application-resource-group protection-group
```

application-resource-group

Specifies the name of an application resource group. You can specify more than one resource group in a comma-separated list.

protection-group

Specifies the name of the protection group.

The `geopg add-resource-group` command adds an application resource group to a protection group on the local cluster. If the partner cluster contains a protection group of the same name, the command propagates the new configuration information to the partner cluster.

For information about the names and values that are supported by disaster recovery framework software, see [Appendix B, “Legal Names and Values of Disaster Recovery Framework Entities,”](#) in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

After the application resource group is added to the protection group, the application resource group is managed as an entity of the protection group. The application resource group is now affected by protection group operations such as start, stop, switchover, and takeover.

- 6. If necessary, bring online the application resource group.**

```
# clresourcegroup online application-resource-group
```

Example 1 Adding an Application Resource Group to an Oracle Solaris ZFS Snapshot Protection Group

The following example adds two application resource groups, `apprg1` and `apprg2`, to the `zfsnap-pg` protection group.

```
# geopg add-resource-group apprg1,apprg2 zfsnap-pg
```

Troubleshooting If the add operation is unsuccessful on the local cluster, the configuration of the protection group is not modified. Otherwise, the configuration status is set to `OK` on the local cluster.

If the configuration status is `OK` on the local cluster but the add operation is unsuccessful on the partner cluster, the configuration status is set to `Error` on the partner cluster.

◆◆◆ 2 CHAPTER 2

Administering Oracle Solaris ZFS Snapshot Protection Groups

This chapter contains information about more administration tasks on Oracle Solaris ZFS snapshot replication protection groups.

The chapter contains the following sections:

- [“Replicating an Oracle Solaris ZFS Snapshot Protection Group Configuration to a Partner Cluster” on page 55](#)
- [“Administering Oracle Solaris ZFS Snapshot Replication Components” on page 59](#)
- [“Removing Oracle Solaris ZFS Snapshot Application Resource Groups” on page 67](#)
- [“Checking the Runtime Status of Oracle Solaris ZFS Snapshot Remote Replication” on page 72](#)

Replicating an Oracle Solaris ZFS Snapshot Protection Group Configuration to a Partner Cluster

After you have configured a protection group on the primary cluster, you can replicate the configuration of the protection group to the secondary cluster. If there are any application resource groups in the protection group on the primary, such resource groups must already exist on the secondary cluster as well. Replicating a protection group configuration on to a cluster will also set up the replication component infrastructure on the local cluster. The replication component infrastructure contains the infrastructure and replication groups and resources.

▼ How to Replicate the Oracle Solaris ZFS Snapshot Protection Group Configuration to a Partner Cluster

Note - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, then click the partnership name. In the Protection Groups section, click Get Protection Groups and select the protection group to replicate. For more information about Oracle Solaris Cluster Manager, see [Chapter 12, “Using the Oracle Solaris Cluster Manager Browser Interface”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

Before You Begin Before you replicate the configuration of an Oracle Solaris ZFS snapshot protection group to a partner cluster, ensure that the following conditions are met:

- You have reviewed the information in [“Planning Remote Replication Using Oracle Solaris ZFS Snapshot”](#) on page 24.
- You have performed the prerequisites mentioned in [“Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot”](#) on page 32.
- The protection group is defined on the remote cluster, not on the local cluster.
- The application resource groups in the protection group configuration on the remote cluster exist on the local cluster.
- Application resource groups are in unmanaged state on the local cluster.

Perform this procedure from one node on the local cluster.

1. Assume the root role or assume a role that is assigned the Geo Management RBAC rights profile.

For more information about RBAC, see [“Securing Disaster Recovery Framework Software”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

Note - If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwX:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management RBAC rights profile and Oracle Solaris ZFS snapshot software.

2. **Ensure that the `Auto_start_on_new_cluster` property of the application resource group is set to `False`.**

```
# clresourcegroup show -p Auto_start_on_new_cluster resource-group
```

If necessary, change the property value to `False`.

```
# clresourcegroup set -p Auto_start_on_new_cluster=False resource-group
```

3. **If either partner is a zone cluster, configure an Oracle Solaris Cluster private string in the global zone of the local partner to store the local replication user's SSH passphrase.**

For example, Suppose that the local partner is a global zone and the remote partner is a zone cluster. The replication component is `repcom1`. Type the following in one node of the local partner to configure the Oracle Solaris Cluster private string:

```
$ clps create -b global:repcom1:local_service_passphrase \  
global:repcom1:local_service_passphrase  
<Enter SSH passphrase at prompt>
```

For example, if the local partner is a zone cluster `zc1` and replication component is `repcom1`, type the following in the global zone of one node of the local zone cluster partner to configure the Oracle Solaris Cluster private string:

```
$ clps create -b zc1:repcom1:local_service_passphrase \  
zc1:repcom1:local_service_passphrase  
<Enter SSH passphrase at prompt>
```

4. **Replicate the protection group configuration to the local partner cluster.**

```
newyork-node-1# geopg get -s partnership protection-group
```

partnership

Specifies the name of the partnership from which the protection group configuration information is retrieved.

protection-group

Specifies the name of the protection group.

If no protection group is specified, then all protection groups that exist in the specified partnership on the remote partner are created on the local cluster.

The command retrieves the configuration information of the protection group from the remote cluster and creates the protection group on the local cluster.

Note - The `geopg get` command replicates disaster recovery framework related entities. To replicate Oracle Solaris Cluster resource groups, resource types, and resources, use the `cluster export -t rg,rt,rs` command to generate an XML cluster configuration file and modify the XML file for the expected configuration on the secondary cluster. Run the `cluster create` command with the `-a` option to apply the configuration updates. For more information, see [“How to Configure Oracle Solaris Cluster Software on All Nodes \(XML\)”](#) in *Installing and Configuring an Oracle Solaris Cluster 4.4 Environment* and the `cluster(8CL)` and `cluster(8CL)` man pages.

5. If local partner is a zone cluster, configure a logical hostname resource and resource group in the global zone of the local partner to host the replication hostname. The names of the resource and resource group are not restricted to any specific format. After configuring the logical hostname resource and resource group, perform the following actions:

- Add a strong positive affinity from the logical hostname resource group to the zpool's infrastructure resource group.

Note - Setting the strong positive resource group affinity will print a warning message, if the logical hostname resource group has the `Auto_start_on_new_cluster=TRUE` property while the zpool's infrastructure resource group has `Auto_start_on_new_cluster=FALSE`. This is allowed, since the disaster recovery framework software will bring up the zpool's infrastructure resource group when required, thereby also bringing up the logical hostname resource group due to the affinity.

- Add a offline-restart resource dependency from the logical hostname resource to the zpool's infrastructure storage `HAStoragePlus` resource.
- Ensure that `Auto_start_on_new_cluster` is `TRUE` on the logical hostname resource group. This property is `TRUE` by default. In case the property is `FALSE`, set it to `TRUE`.

For example, suppose the local partner is a zone cluster `zc1` and local replication hostname is `newyork-lh`. The zpool infrastructure resource group in `zc1` is `pg1-targpool1-infr-rg`. The storage resource is `pg1-targpool1-stor-rs`. Type the following commands in the global zone of one node of the local partner `zc1`:

```
$ clrg create newyork-lh-rg
$ clrslh create -g newyork-lh-rg -h newyork-lh newyork-lh-rs
$ clrg manage newyork-lh-rg
$ clrg set -p RG_affinities==zc1:pg1-targpool1-infr-rg newyork-lh-rg
(C538594) WARNING: resource group global:newyork-lh-rg has a strong positive affinity
on
```

```

resource group zc1:pg1-targpool1-infr-rg with Auto_start_on_new_cluster=FALSE;
global:newyork-lh-rg will be forced to remain offline until its strong affinities are
satisfied.
$ clrs set -p Resource_dependencies_offline_restart=zc1:pg1-targpool1-stor-rs newyork-
lh-rs
$ clrg show -p Auto_start_on_new_cluster newyork-lh-rg
=== Resource Groups and Resources ===
Resource Group:          newyork-lh-rg
Auto_start_on_new_cluster:  True
*****

```

If the property is not True, type the following command :

```
$ clrg set -p Auto_start_on_new_cluster=True paris-lh-rg
```

Troubleshooting If the validation is successful, the configuration status is set to OK, and the protection group is created on the local cluster. This protection group contains a remote replication component and application group that are configured almost identically to the replication component and application group on the remote cluster.

If the validation fails, the protection group is not created on the local cluster. Fix the cause of the error, and replicate it again.

If you have difficulties replicating the protection group configuration, see [“Debugging an Oracle Solaris ZFS Snapshot Protection Group” on page 40](#).

Administering Oracle Solaris ZFS Snapshot Replication Components

This section provides the following information for administering replication components in an Oracle Solaris ZFS snapshot protection group:

- [“How to Modify an Oracle Solaris ZFS Snapshot Replication Component” on page 59](#)
- [“How to Remove a Replication Component From an Oracle Solaris ZFS Snapshot Protection Group” on page 63](#)

▼ How to Modify an Oracle Solaris ZFS Snapshot Replication Component

Before You Begin Ensure that the following conditions are met:

- The disaster recovery framework software is installed on the primary and secondary clusters.
- You have reviewed the information in [“Planning Remote Replication Using Oracle Solaris ZFS Snapshot” on page 24](#).
- You have performed the prerequisites mentioned in [“Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot” on page 32](#)
- The local cluster is a member of a partnership.
- The protection group and the replication component already exist on the local cluster.

1. Assume the root role or assume a role that is assigned the Geo Management RBAC rights profile.

For more information about RBAC, see [“Securing Disaster Recovery Framework Software” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Note - If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management RBAC rights profile and Oracle Solaris ZFS snapshot software.

2. The script-based plug-in evaluation rules for the replication component is correct in the file that is already associated with the protection group. The protection group configuration output of `geopg show` will display which file is associated with the protection group.

For more information about configuration files, see [“configuration_file Property” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

3. Update the ZFS snapshots configuration file with the changed information for an existing replication component. For the parameters that you do not want to modify, specify the same values that exist in the replication component.

Note - You must specify existing values for unchanged parameters and new values for the parameters to be modified. Hence, it is recommended to save the configuration file after you create the replication component, so that you can edit required values when you want to modify the replication component.

- a. **If you have a saved copy of the configuration file used to create the replication component, use that file. Otherwise, make a copy of the `/opt/ORCLscgprepzfssnap/etc/zfs_snap_geo_config` replication configuration file to a different location, such as `/var/tmp/geo/zfs_snapshot/zfs_snap_geo_config`.**

```
# cp /opt/ORCLscgprepzfssnap/etc/zfs_snap_geo_config /var/tmp/geo/zfs_snapshot
```

- b. **Edit this replication configuration file with updates for the existing replication component. Fill in the existing values for unchanged parameters and new values for the parameters that require to be modified.**

For example, if you want to modify the `REPLICATION_INTERVAL` to 180 and `NUM_OF_SNAPSHOTS_TO_STORE` to 1 for an existing replication component `repcom1` in a protection group `pg1` configured between clusters `paris` and `newyork`. If the replication configuration file to use for `repcom1` is `/var/tmp/geo/zfs_snapshot/zfs_snap_geo_config`, then set these new values in that file, keeping all other parameters unchanged.

```
PS=paris-newyork
PG=pg1
REPCOMP=repcom1
REPRS=repcom1-repstatus-rs
REPRG=pg1-repstatus-rg
DESC="Protect app1-rg1 using ZFS snapshot replication"
APPRG=app1-rg1
CONFIGFILE=/var/tmp/geo/zfs_snapshot/sbp_conf
LOCAL_REP_USER=zfsuser1
REMOTE_REP_USER=zfsuser2
LOCAL_ZPOOL_RS=par-app1-haspl
REMOTE_ZPOOL_RS=ny-app1-haspl
LOCAL_LH=paris-lh
REMOTE_LH=newyork-lh
LOCAL_DATASET=srcpool1/app1-ds1
REMOTE_DATASET=targpool1/app1-ds1-copy
REPLICATION_INTERVAL=180
NUM_OF_SNAPSHOTS_TO_STORE=1
REPLICATION_STREAM_PACKAGE=false
SEND_PROPERTIES=true
INTERMEDIARY_SNAPSHOTS=false
RECURSIVE=true
MODIFY_PASSPHRASE=false
```

4. **On the node of the local partner having the changed replication configuration file, run the `zfs_snap_geo_register` script to modify the replication component.**

Use the `zfs_snap_geo_register` script with the modified configuration file.

```
# /opt/ORCLscgrepzfssnap/util/zfs_snap_geo_register -f /var/tmp/geo/zfs_snapshot/
zfs_snap_geo_config
```

The command modifies the replication component in a protection group on the local cluster. If the partner cluster contains a protection group with the same name, the command also propagates the new configuration to the partner cluster.

You can perform this procedure from either cluster. However, if you are using a saved configuration file that was used to create replication component from the primary cluster, perform this procedure on the primary cluster since the values in the file will be appropriate from the primary cluster.

5. Check that the new parameters reflect in the configuration of the replication component on each partner cluster.

For example, you can type the following command on one node of paris cluster and one node of newyork cluster:

```
# geopg show pg1
.....
    SBP Script-based plug-in configurations:
        repcom1
        .....
        repcom1_create_config_args :
        .....,REPLICATION_INTERVAL=180,NUM_OF_SNAPSHOTS_TO_STORE=1,.....
```

You can also check the relevant extension properties of the replication agent resource associated with the replication component to check that the new values have been set correctly. For information about extension properties of replication agent resource type ORCL.ZFSSnapRepAgent, see [Appendix A, “Disaster Recovery Framework Properties for Oracle Solaris ZFS Snapshot”](#).

For example, you can type the following command on one node of paris cluster and one node of newyork cluster to check if the configuration is updated.

```
# /usr/cluster/bin/clresource show -p REPLICATION_INTERVAL -p NUM_OF_SNAPSHOTS_TO_STORE
repcom1-snap-rs
=== Resources ===
Resource:          repcom1-snap-rs
  --- Standard and extension properties ---
  Num_of_snapshots_to_store:      1
  Class:                          extension
  Description:                    The number of replication snapshots to keep.
  Per-node:                       False

  Type:                            int
  Replication_interval:           180
```

Class:	extension
Description:	The interval in seconds between ZFS snapshot replication cycles.
Per-node:	False
Type:	int

Note - If you only modify parameters that correspond to those extension properties of replication agent resource type ORCL.ZFSSnapRepAgent that can be tuned anytime, then such modification attempt will not disable the replication agent resource when modifying such extension properties. But if you modify any parameters that correspond to extension properties that can be tuned when the resource is disabled, then such modification attempt will disable the replication agent resource, modify its properties, and then bring it back online if it was online earlier.

▼ How to Remove a Replication Component From an Oracle Solaris ZFS Snapshot Protection Group

Note - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. First you must remove affinity and dependency from the replication logical hostname resource group and resource in case of zone cluster, as described below. Then click Partnerships, click the partnership name, click the protection group name, in the Data Replication Components section highlight the replication component name, and click Remove. After this is done, remove the Oracle Solaris Cluster private strings as described below in case either partner is a zone cluster. For more information about Oracle Solaris Cluster Manager, see [Chapter 12, “Using the Oracle Solaris Cluster Manager Browser Interface” in *Administering an Oracle Solaris Cluster 4.4 Configuration*](#).

Before You Begin Ensure that the following conditions are met:

- The protection group is defined on the local cluster.
- The protection group is deactivated on both partner clusters.
- The replication component is managed by the protection group.

1. **Assume the root role or assume a role that is assigned the Geo Management RBAC rights profile.**

For more information about RBAC, see [“Securing Disaster Recovery Framework Software” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Note - If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the `root` role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rxw:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management RBAC rights profile and Oracle Solaris ZFS snapshot software.

- 2. If local partner is a zone cluster, remove the resource group affinity and resource dependency from logical hostname resource group and resource that is hosting the replication hostname in the global zone. You would have set the affinity and dependency when creating the replication setup. Optionally, you can remove the logical hostname resource group and resource.**

For example, suppose the logical hostname resource group in global zone is `paris-lh-rg`, and the resource is `paris-lh-rs`.

Type the following commands in the global zone of one node of the local partner to only remove the affinity and dependency:

```
# clrg set -p RG_affinities==++paris:pg1-srcpool1-infr-rg paris-lh-rg
# clrs set -p Resource_dependencies_offline_restart==paris:pg1-srcpool1-stor-rs paris-lh-rs
```

Type the following commands in the global zone of one node of the local partner to remove the resource group and resource:

```
# clrs disable paris-lh-rs
# clrs delete paris-lh-rs
# clrg delete paris-lh-rg
```

- 3. If the protection group and the replication component exists on the remote partner, then the replication configuration related to this replication component on the remote partner will be removed automatically.**

If a remote partner is a zone cluster, remove the resource group affinity and resource dependency from logical hostname resource group and resource that is hosting the replication hostname in the global zone of the remote partner. You would have set the affinity and dependency when creating the replication setup. Optionally, you can remove the logical hostname resource group and resource.

For example, suppose the logical hostname resource group in global zone is `newyork-lh-rg`, and the resource is `newyork-lh-rs`.

Type the following commands in the global zone of one node of the remote partner to only remove the affinity and dependency:

```
# clrg set -p RG_affinities=+++newyork:pg1-targpool1-infr-rg newyork-lh-rg
# clrs set -p Resource_dependencies_offline_restart=newyork:pg1-targpool1-stor-rs
  newyork-lh-rs
```

Type the following commands in the global zone of one node of the remote partner to remove the resource group and resource:

```
# clrs disable newyork-lh-rs
# clrs delete newyork-lh-rs
# clrg delete newyork-lh-rg
```

4. Remove the remote replication component.

Type the following command in one node of the local partner cluster.

```
# geopg remove-replication-component zfs-snap-replication-component zfs-snap-protection-group
```

zfs-snap-replication-component

Specifies the name of the replication component.

zfs-snap-protection-group

Specifies the name of the protection group.

If the replication component exists on the remote partner, then this command removes the replication component configuration from the remote partner automatically. This happens in an asynchronous manner.

5. If either partner is a zone cluster, you must remove the Oracle Solaris Cluster private string in the global zone of each partner that was created to store the SSH passphrase used by the replication setup.

For example, if *paris* is a global cluster and the other partner *newyork* is a zone cluster, and the replication component is *repcom1*, type the following command in one node of *paris* to remove the private string *global:repcom1:local_service_passphrase*.

```
$ clps delete global:repcom1:local_service_passphrase
```

Type the following command in the global zone on one node of *newyork* to remove the private string *newyork:repcom1:local_service_passphrase*.

```
$ clps delete newyork:repcom1:local_service_passphrase
```

6. On each zone cluster partner, if no other existing replication components in the protection group use the zpool that was used by the removed replication

component, which means that the only SUNW.HAStoragePlus resource managing the zpool is in the application resource group, then you may delete the zpool's device group in the global zone of a zone cluster partner. That zpool device group would have been created by the replication infrastructure when you had created the replication component that you just removed. The name of the such a zpool device group is same as the name of the zpool.

For example, type the following command in the global zone of a zone cluster partner newyork to delete the device group for the zpool targpool1:

```
$ cldg delete targpool1
```

You are not required to perform this action as the presence of the zpool device group does not hamper any supported action on a zpool in the Oracle Solaris Cluster environment.

Changing Oracle Solaris ZFS Snapshot Application Resource Groups Between Failover and Scalable

When the application resource group is changed from scalable to failover or from failover to scalable, you must remove the replication component from the protection group then re-add it to have the correct infrastructure resource group and resources created for the application resource group.

The Oracle Solaris ZFS snapshot infrastructure resource group name is dependent on `RG_mode` of the application resource group where the `HAStoragePlus` resource managing the zpool being replicated resides. If the application resource group is of failover type, the infrastructure resource group name is `<pg>-<AppRGName>-infr-rg`. If the application resource group is of scalable type, the infrastructure resource group name is `<pg>-<zpool>-infr-rg`.

When the application is changed from scalable to failover or from failover to scalable, you must remove the replication component from the protection group then re-add it to have the correct infrastructure resource group and resources created for the application resource group.

▼ How to Change the Resource Type of an Application Resource Group

The `rg_mode` can only be set at creation time. In order to change the application resource group type, you must delete the current resource group, and re-create it with the new type. Use this procedure to change an application resource group type.

1. **Issue the `geopg show <pg>` command and have the output available to re-enter this info when the replication component is re-added to the protection group.**
2. **Remove the replication component from the protection group.**
See [“How to Remove an Application Resource Group From an Oracle Solaris ZFS Snapshot Protection Group”](#) on page 67.
3. **Create or re-create the application resource group with the new `rg_mode` you require. If this has already been done go to the next step.**
4. **Re-add the replication component to the protection group.**
Refer to the output from Step 1. If the application resource group name changed, use the new name.
See [“How to Add a Replication Component to an Oracle Solaris ZFS snapshot Protection Group”](#) on page 44.

Removing Oracle Solaris ZFS Snapshot Application Resource Groups

To make an application highly available, the application must be managed as a resource in an application resource group. This section contains information about removing an application resource group from an Oracle Solaris ZFS snapshot protection group.

▼ How to Remove an Application Resource Group From an Oracle Solaris ZFS Snapshot Protection Group

In disaster recovery framework ZFS snapshot replication feature, application resource group in a protection group is closely bound to the replication components that is associated to the application resource group. You must remove the application resource group and its associated replication component to completely remove the application resource group from the protection group's control. You could remove the replication component first and then the application resource group, or vice versa.

Note that you can remove any such configuration from the protection group only when the protection group is deactivated, which in turn implies that the application resource group is in unmanaged state.

Note - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name, click the protection group name, highlight the resource group name, and click Remove. For more information about Oracle Solaris Cluster Manager, see [Chapter 12, “Using the Oracle Solaris Cluster Manager Browser Interface” in *Administering an Oracle Solaris Cluster 4.4 Configuration*](#).

Before You Begin Ensure that the following conditions are met:

- The protection group is defined on the local cluster.
- The resource group to be removed is part of the application resource groups of the protection group.
- The protection group is deactivated on both partner clusters.

1. Assume the root role or assume a role that is assigned the Geo Management RBAC rights profile.

For more information about RBAC, see [“Securing Disaster Recovery Framework Software” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Note - If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwx:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management RBAC rights profile and Oracle Solaris ZFS snapshot software.

2. Remove the application resource group from the protection group.

Type the following command in one node of the local partner cluster.

```
# geopg remove-resource-group application-resource-group protection-group
```

application-resource-group

Specifies the name of an application resource group. You can specify more than one resource group in a comma-separated list.

protection-group

Specifies the name of the protection group.

The command removes an application resource group from a protection group on the local cluster. If the partner cluster contains a protection group of the same name, the command also removes the application resource group from the protection group on the partner cluster.

- 3. If one or more replication components associated with the application still exists in the protection group, remove such replication components as described in [“How to Remove a Replication Component From an Oracle Solaris ZFS Snapshot Protection Group”](#) on page 63.**

Example 2 Removing an Oracle Solaris ZFS Snapshot Application Resource Group From a Protection Group

The following example removes two application resource groups, `apprg1` and `apprg2`, from the `zfsnap-pg` protection group.

```
# geopg remove-resource-group apprg1,apprg2 zfsnap-pg
```

Troubleshooting If the remove operation is unsuccessful on the local cluster, the configuration of the protection group is not modified. Otherwise, the configuration status is set to OK on the local cluster.

If the configuration status is OK on the local cluster but the remove operation is unsuccessful on the partner cluster, the configuration status is set to Error on the partner cluster.

Deleting Oracle Solaris ZFS Snapshot Protection Group From a Partner Cluster

This section describes the procedure to delete the protection group from a partner cluster.

▼ How to Remove a Oracle Solaris ZFS Snapshot Protection Group From a Partner Cluster

Note - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. First you must remove affinity and dependency from the replication logical hostname resource group and resource in case of zone cluster, as described below. Then click Partnerships, click the partnership name, click the protection group name, in the Data Replication Components section highlight the replication component name, and click Remove. After this is done, remove the Oracle Solaris Cluster private strings as described below in case either partner is a zone cluster. For more information about Oracle Solaris Cluster Manager, see [Chapter 12, “Using the Oracle Solaris Cluster Manager Browser Interface” in *Administering an Oracle Solaris Cluster 4.4 Configuration*](#).

Before You Begin Ensure that the following conditions are met:

- The protection group is defined on the local cluster.
- The protection group is deactivated on the partner cluster from which you want to delete it.
- You have performed the prerequisites mentioned in [“Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot” on page 32](#).

1. Assume the root role or assume a role that is assigned the Geo Management RBAC rights profile.

For more information about RBAC, see [“Securing Disaster Recovery Framework Software” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Note - If you use a role with Geo Management RBAC rights, ensure that the `/var/cluster/geo` ACLs are correct on each node of both partner clusters. If necessary, assume the root role on the cluster node and set the correct ACLs.

```
# chmod A+user:username:rwX:allow /var/cluster/geo
```

The `/var/cluster/geo` directory must have the correct access control lists (ACL) applied for compatibility between the Geo Management RBAC rights profile and Oracle Solaris ZFS snapshot software.

2. If local partner is a zone cluster, for each replication component in the protection group remove the resource group affinity and resource dependency from logical hostname resource group and resource that is hosting the replication hostname in the global zone. You would have set the affinity and dependency when creating the replication setup. Optionally you can remove the replication

logical hostname resource group and resource for each replication component in the protection group.

For example, suppose the local partner `paris` is a zone cluster and the logical hostname resource group in global zone is `paris-lh-rg`, and the resource is `paris-lh-rs`. Type the following commands in the global zone on one node of `paris` to remove the resource group and resource:

```
$ clrs disable paris-lh-rs
$ clrs delete paris-lh-rs
$ clrg delete paris-lh-rg
```

Type the following commands in the global zone of one node of `paris` to only remove the affinity and dependency:

```
$ clrg set -p RG_affinities=+++paris:pg1-srcpool1-infr-rg paris-lh-rg
$ clrs set -p Resource_dependencies_offline_restart=paris:pg1-srcpool1-stor-rs paris-lh-rs
```

3. Remove the protection group from local cluster.

Type the following command in one node of the local partner cluster:

```
# geopg delete protection-group
```

where `protection-group` is the name of the protection group.

4. If either partner is a zone cluster, you can remove the Oracle Solaris Cluster private string for each replication component in the global zone of local partner that was created to store the SSH passphrase used by the replication setup.

For example, if `paris` is a global cluster and the other partner `newyork` is a zone cluster, and a replication component is `repcom1`, type the following command in one node of `paris` to remove the private string `global:repcom1:local_service_passphrase`:

```
$ clps delete global:repcom1:local_service_passphrase
```

Suppose if `paris` is a zone cluster, and a replication component is `repcom1`, type the following command in the global zone on one node of `paris` to remove the private string `paris:repcom1:local_service_passphrase`.

```
$ clps delete paris:repcom1:local_service_passphrase
```

5. (Optional) If the only `SUNW.HASStoragePlus` resource managing the zpool is in the application resource group, then you may delete the zpool's device group in the global zone of a zone cluster partner. That zpool device group would have been created by the replication infrastructure when you had created the replication

component that you just removed. The name of the such a zpool device group is same as the name of the zpool.

For example, type the following command in the global zone of a zone cluster partner paris to delete the device group for the zpool `srcpool1`:

```
$ cldg delete srcpool1
```

You are not required to perform this action as the presence of the zpool device group does not hamper any supported action on a zpool in the Oracle Solaris Cluster environment.

Checking the Runtime Status of Oracle Solaris ZFS Snapshot Remote Replication

The disaster recovery framework software internally creates and maintains one replication status resource group for each protection group. The name of the replication status resource group is specified by the user in the configuration file as described in [“How to Create and Configure an Oracle Solaris ZFS Snapshot Protection Group”](#) on page 34.

You can obtain an overall view of the status of replication as well as a more detailed runtime status of the replication status resource groups. The following sections describe the procedures for checking each status:

- [“Overview of Displaying an Oracle Solaris ZFS Snapshot Runtime Status”](#) on page 72
- [“How to Check the Runtime Status of Oracle Solaris ZFS Snapshot Replication”](#) on page 73
- [“Oracle Solaris ZFS Snapshot Replication Resource Runtime Status and Status Messages”](#) on page 74

Overview of Displaying an Oracle Solaris ZFS Snapshot Runtime Status

The status of all the resources under a protection group are aggregated in the replication status reported by the protection group. This replication status is displayed in the protection group status. For more information about the status of protection groups, refer to [“Monitoring the Runtime Status of the Disaster Recovery Framework”](#) in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

If you add an Oracle Solaris ZFS snapshot component to a protection group, disaster recovery framework software creates a replication status resource for the replication component. This resource monitors the status of replication for its replication component.

You can monitor the status of replication of this replication component by checking the Status and Status Message of this resource. Use the `clresource status` command to display resource status and the status message.

▼ How to Check the Runtime Status of Oracle Solaris ZFS Snapshot Replication

Note - You can also accomplish this procedure by using the Oracle Solaris Cluster Manager browser interface. Click Partnerships, click the partnership name, and click the protection group name. For more information about Oracle Solaris Cluster Manager, see [Chapter 12, “Using the Oracle Solaris Cluster Manager Browser Interface”](#) in *Administering an Oracle Solaris Cluster 4.4 Configuration*.

1. Access a node of the cluster where the protection group has been defined.

You must be assigned the Basic Solaris User RBAC rights profile to complete this procedure. For more information about RBAC, see [“Securing Disaster Recovery Framework Software”](#) in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

2. Check the runtime status of Geographic Edition components.

```
# geoadm status
```

Refer to the Protection Group section of the output for replication information. The information that is displayed by this command includes the following:

- Whether the local cluster is enabled for partnership participation.
- Whether the local cluster is involved in a partnership.
- Status of the heartbeat configuration.
- Status of the defined protection groups, which also shows the replication status of each protection group.
- Status of current transactions.

3. Check the runtime status of replication for each Oracle Solaris ZFS snapshot replication component by checking the status of the associated replication status resource.

```
# clresourcegroup status zfs-snap-replication-resource-group
```

```
# clresource status zfs-snap-replication-resource
```

Refer to the Status and Status Message fields for the replication status resource associated with the replication component you want to check. See [“Oracle Solaris ZFS Snapshot Replication Resource Runtime Status and Status Messages”](#) on page 74 for a list of possible status values and status messages.

Oracle Solaris ZFS Snapshot Replication Resource Runtime Status and Status Messages

The following table lists the Status and Status Message values that are returned by the `clresource status` command when the State of the Oracle Solaris ZFS snapshot replication status resource is not OffLine.

TABLE 4 Status and Status Messages of an Online Oracle Solaris ZFS snapshot Replication Status Resource Group

Situation	Status	Status Message
The specified replication hostname is unreachable	UNKNOWN	Host <i>hostname</i> is unreachable
The snapshot replication daemon of the resource is not running	FAULTED	The snapshot replication daemon is not running
The SSH agent for the resource is not running	FAULTED	SSH agent is not running
The ZFS dataset configured for replication on local cluster is not found	FAULTED	Local ZFS dataset <i>local_dataset</i> is not found
The ZFS command to list all ZFS datasets on the specified host failed	FAULTED	Failed to get the list of all ZFS datasets on the host <i>host</i>
The ZFS command to list all snapshots of a ZFS dataset on a specified host failed	FAULTED	Failed to get the list of ZFS snapshots for the ZFS dataset <i>dataset</i> on the host <i>host</i>
The ZFS command to get the GUID property of a ZFS snapshot failed	FAULTED	Failed to get the GUID of the ZFS snapshot <i>snapshot_name</i> on the host <i>host</i>
The ZFS command to create a recursive/non-recursive snapshot failed	FAULTED	Failed to create <i>recursive/non-recursive</i> ZFS snapshot <i>snapshot_name</i> on the host <i>local_host</i>
The ZFS command to roll back a ZFS dataset to an earlier snapshot failed	FAULTED	Failed to roll back the ZFS dataset <i>dataset</i> on the host <i>host</i> to the ZFS snapshot <i>snapshot_name</i>
The ZFS command to destroy a ZFS snapshot failed	FAULTED	Failed to destroy the ZFS snapshot <i>snapshot_name</i> of the ZFS dataset <i>dataset</i> on the host <i>host</i>
ZFS send/receive command pipeline failed to replicate a snapshot	FAULTED	Failed to replicate the ZFS snapshot <i>snapshot_name</i> from the source dataset

Situation	Status	Status Message
		<i>source_dataset</i> on the host <i>local_host</i> to the target dataset <i>target_dataset</i> on the host <i>remote_host</i>
Replication agent daemon is attempting one round of replication	OK	Attempting replication
The ZFS command to get the dataset hierarchy of the specified dataset failed	FAULTED	Failed to get the dataset hierarchy for the local dataset <i>local_dataset</i>
Replication stream package cannot be used to perform replication, as a recursive send is not possible when datasets in a hierarchy are not at the same replicated snapshot level	FAULTED	Cannot replicate the dataset hierarchy using a single ZFS replication stream package, because the latest replicated snapshots on the datasets in the hierarchy have different names. Use the RECURSIVE parameter instead of the REPLICATION_STREAM_PACKAGE parameter.
The specified snapshot was replicated successfully according to the replication parameters	OK	Synchronized to snapshot <i>snapshot_name</i>
The resource could not get the SSH passphrase to use	FAULTED	Failed to get the SSH passphrase for replication user <i>replication_user</i> from the Oracle Solaris Cluster private string " <i>OSC_private_string</i> "
The resource failed to get the zpools property value of the specified SUNW.HASStoragePlus resource	FAULTED	Failed to query Zpools property of the HASStoragePlus resource <i>HASStoragePlus_resource_name</i> in the zone <i>remote_zone</i> of the remote host <i>remote_host</i>
The infrastructure SUNW.HASStoragePlus resource on the remote partner does not manage the remote zpool	FAULTED	The remote HASStoragePlus resource <i>HASStoragePlus_resource_name</i> in the zone <i>remote_zone</i> of the remote host <i>remote_host</i> does not manage the zpool <i>zpool_name</i>
The infrastructure SUNW.HASStoragePlus resource, that must have been created automatically by the replication setup software, does not exist on the local cluster	FAULTED	The infrastructure HASStoragePlus resource <i>SUNW.HASStoragePlus_resource_name</i> does not exist on the local cluster
The infrastructure SUNW.HASStoragePlus resource, that manages the zpool on the local cluster, is not online	DEGRADED	The infrastructure HASStoragePlus resource <i>SUNW.HASStoragePlus_resource_name</i> is not online on the local cluster
The associated snapshot replication agent resource of type ORCL.ZFSSnapRepAgent, that should be created automatically by the replication setup software, does not exist on the local cluster.	FAULTED	The snapshot replication agent resource <i>resource_name</i> does not exist on the local cluster
The associated snapshot replication agent resource of type ORCL.ZFSSnapRepAgent is not online on the local cluster.	DEGRADED	The snapshot replication agent resource <i>resource_name</i> is not online on the local cluster
The status resource could not get the replication status from the snapshot replication agent on the remote primary	DEGRADED	Failed to obtain replication status from the snapshot replication agent resource <i>resource_name</i> in the zone <i>remote_zone</i> on the remote host <i>remote_host</i>

Situation	Status	Status Message
The SSH information stored by this resource is missing, so the resource cannot connect to remote host for its status query	FAULTED	Cannot find the SSH environment file <i>file</i>

Examine the resource log file, console and syslog messages, and the trace log for more information about the failures. The ZFS commands executed by the replication module can be seen in the resource log and the trace log. For more information about such failure messages from ZFS commands, see [Chapter 8, “Working With Oracle Solaris ZFS Snapshots and Clones” in *Managing ZFS File Systems in Oracle Solaris 11.4*](#) and the `zfs(8)` man page.

Typically, the ZFS command failures might occur due to the absence of the required ZFS dataset permissions for the replication user. For information about the required ZFS permissions, see [“Prerequisites for Configuring Remote Replication Using Oracle Solaris ZFS Snapshot” on page 32](#).

For more information about the `clresource` command, see the `clresource(8CL)` man page.

Migrating Services That Use Oracle Solaris ZFS Snapshot Remote Replication

This chapter provides information about migrating services for maintenance or as a result of cluster failure. This chapter contains the following sections:

- [“Switchover Services to a Partner Cluster in a Planned Manner” on page 77](#)
- [“Recovering Services to a Cluster on a System That Uses Oracle Solaris ZFS Snapshot Replication” on page 80](#)
- [“Recovering From an Oracle Solaris ZFS snapshot Remote Replication Error” on page 87](#)

Switchover Services to a Partner Cluster in a Planned Manner

This section describes how to perform a protection group switchover in order to migrate services to a partner cluster in a planned manner.

▼ How to Perform a Switchover for a Oracle Solaris ZFS Snapshot Protection Group

Use the standard protection group switchover procedure to perform switchover of a ZFS snapshot replication protection group to migrate services in a planned manner. For more information, see [“Migrating Replication Services by Switching Over Protection Groups” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Before You Begin Ensure that the following conditions are met before you attempt a switchover:

- Disaster Recovery Framework infrastructure is enabled on both partner clusters, and the partnership status is OK.
- Protection group is activated on both partner clusters.
- Protection group status and data replication status is OK.
- Protection group has the primary role on one partner and the secondary role on the other partner.

This procedure uses the example names `paris` for the current primary cluster and `newyork` for the current secondary cluster. The goal is to switchover the protection group to make `newyork` the new primary.

1. **Type this command in one node of `paris` or in one node of `newyork` to do the switchover to `newyork`.**

```
# geopg switchover [-f] -m newyork protection-group
```

After the switchover is complete, ensure that the protection group and replication status is OK on both partners.

2. **Verify that the protection group is now primary on `newyork` and secondary on `paris`, and that the state for Data replication and Resource groups is OK on both clusters.**

```
# geopg status protection-group
```

3. **Check the runtime status of the application resource group and replication status resources associated with the protection group.**

```
# clresourcegroup status application-rg-in-pg
# clresource status -g application-rg-in-pg
# clresourcegroup status replication-status-rg-for-pg
# clresource status replication-status-rs
```

Refer to the status and status message fields for the replication status resource of each remote replication component in the protection group. For more information about the runtime status of replication, see [“Checking the Runtime Status of Oracle Solaris ZFS Snapshot Remote Replication” on page 72](#).

Migrating Services That use ZFS Snapshot Replication with a Takeover

This section describes how to perform a protection group takeover in order to migrate services to a partner cluster. You perform a takeover when applications need to be brought online on the existing secondary partner cluster regardless of whether the data is completely synchronized between the primary and secondary ZFS datasets. The applications will be started on the existing secondary partner as a result of the takeover if the protection group has been activated on that partner.

▼ How to Perform a Takeover for a Oracle Solaris ZFS Snapshot Protection Group

Use the standard protection group takeover procedure to perform takeover of a ZFS snapshot replication protection group to migrate services to the existing secondary partner. For more information, see [“Forcing a Takeover of a Protection Group” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

Before You Begin Ensure that the following conditions are met before you attempt a takeover:

- Disaster Recovery Framework infrastructure is up and running on the secondary cluster.
- Protection group is configured on the secondary cluster, and its Configuration status is OK

This procedure uses the example names `paris` for the current primary cluster and `newyork` for the current secondary cluster. The goal is to takeover the protection group to make `newyork` the new primary.

1. **Type the following command in one node of `newyork` to do the takeover to `newyork`.**

```
# geopg takeover [-f] protection-group
```

2. **After the takeover is complete, check that the protection group is now primary on `newyork`.**

```
# geopg status protection-group
```

3. **If the disaster recovery framework infrastructure is up and running on the original primary `paris` and the disaster recovery framework infrastructure on `newyork` can reach `paris`, then check on `paris` that it is secondary now and that the protection group is deactivated on `paris`.**

```
#geoadm status protection-group
```

4. **Data replication is deactivated on a takeover. Check that the replication status resources associated with the protection group are offline.**

```
# clresourcegroup status -v replication-status-rg-for-pg  
# clresource status -v replication-status-rs
```

5. **If the protection group is in activated state on the new primary newyork, then check that the application resource group and its resources are online. Otherwise, activate the protection group on newyork to bring the application online. Type the following commands to check the status of the application:**

```
# clresourcegroup status application-rg-in-pg  
# clresource status -g application-rg-in-pg
```

Recovering Services to a Cluster on a System That Uses Oracle Solaris ZFS Snapshot Replication

This section describes the recovery process. It contains the following information:

- [“Overview of Recovering Services” on page 80](#)
- [“How to Perform a Failback-Switchover on a System That Uses Oracle Solaris ZFS Snapshot Replication” on page 81](#)
- [“How to Perform a Failback-Takeover on a System That Uses Oracle Solaris ZFS Snapshot Replication” on page 84](#)

Overview of Recovering Services

After a successful takeover operation, the secondary cluster becomes the primary for the protection group and the services are online on the secondary cluster. After the recovery of the original primary cluster, the services can be brought online again on the original primary by using a process called *failback*.

The disaster recovery framework software supports the following kinds of failback:

- **Failback-switchover.** During a failback-switchover, applications are brought online again on the original primary cluster after the data of the original primary cluster was resynchronized with the data on the secondary cluster.

- **Failback-takeover.** During a failback-takeover, applications are brought online again on the original primary cluster and use the current data on the original primary cluster. Any updates that occurred on the secondary cluster while it was acting as primary are discarded.

If you want to leave the new primary as the primary cluster and the original primary cluster as the secondary after the original primary restarts, you can resynchronize and revalidate the protection group configuration without performing a switchover or takeover.

▼ How to Perform a Failback-Switchover on a System That Uses Oracle Solaris ZFS Snapshot Replication

Use this procedure to restart an application on the original primary cluster, `paris`, after the data on this cluster has been resynchronized with the data on the current primary cluster, `newyork`.

Note - The failback procedures apply only to clusters in a partnership.

Before You Begin Before you perform a failback-switchover, a takeover has occurred on `newyork`. Ensure that the following conditions are met:

- If the original primary cluster has been down, the cluster has been booted and the disaster recovery framework infrastructure is enabled on the cluster. For more information about booting a cluster, see [“Booting a Cluster” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).
- The protection group on the current primary cluster has the `primary` role.
- The protection group on the original primary cluster has either the `primary` role or `secondary` role depending on whether the original primary cluster can be reached during the takeover from the current primary cluster.

This procedure uses the example names `paris` for the original primary cluster and `newyork` for the current primary cluster.

1. **Ensure that the protection group is stopped at the original primary cluster `paris` site.**
 - a. **Determine whether the protection group on the original primary cluster, `paris`, is active.**

```
paris-node-1# geopg status protection-group
```

- b. If the protection group on the original primary cluster is active, stop it.**

```
paris-node-1# geopg stop -e local protection-group
```

- c. Verify that the protection group is stopped.**

```
paris-node-1# geopg status protection-group
```

- 2. Resynchronize the original primary cluster, paris, with the current primary cluster, newyork.**

The paris cluster forfeits its own configuration and replicates the newyork configuration locally. Resynchronize both the partnership and protection group configurations.

- a. On paris, resynchronize the partnership.**

```
paris-node-1# geops update partnership
```

Note - Perform this step only once per partnership even if you are performing a failback-switchover for multiple protection groups in the partnership.

For more information about synchronizing partnerships, see [“Resynchronizing a Partnership” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

- b. On paris, resynchronize the protection group.**

Because the local role of the protection group on newyork is now primary, this step ensures that the role of the protection group on paris becomes secondary.

```
paris-node-1# geopg update protection-group
```

For more information about synchronizing protection groups, see [“Resynchronizing a Protection Group” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

- 3. On paris, validate the cluster configuration for the protection group.**

Ensure that the protection group is not in an error state. A protection group cannot be started when it is in an error state.

```
paris-node-1# geopg validate protection-group
```

For more information, see [“Validating a Protection Group” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

4. On paris, activate the protection group.

Because the protection group on `paris` has a role of secondary, the `geopg start` command does not restart the application on `paris`.

```
paris-node-1# geopg start -e global protection-group
```

```
-e global
```

Specifies the scope of the command. The `global` scope applies the command on both clusters.

```
protection-group
```

Specifies the name of the protection group.

Note - Do not use the `-n` option when performing a failback-switchover. The data must be synchronized from the current primary cluster, `newyork`, to the current secondary cluster, `paris`.

Because the protection group has a role of secondary on `paris`, the data is synchronized from the current primary, `newyork`, to the current secondary, `paris`.

For more information about the `geopg start` command, see [“How to Activate a Protection Group”](#) in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*.

5. Confirm that the data is completely synchronized.

The status of replication progress is reported by the replication status resources for replication components in the protection group. Check such resource status to determine whether the data has been synchronized. Also, the state of the protection group becomes OK.

To confirm whether the state of the protection group on `paris` and `newyork` is OK, use the following command:

```
paris-node-1# geoadm status
newyork-node-1# geoadm status
```

Refer to the `Protection group` section of the output.

Check the status of the replication status resources for the replication components in the protection group to check that the data is completely synchronized.

```
# clresourcegroup status -v replication-status-rg-for-pg
# clresource status -v replication-status-resource
```

6. On both partner clusters, ensure that the protection group is activated.

```
# geopg status protection-group
```

7. **On either cluster, perform a switchover from newyork to paris for the protection group.**

```
# geopg switchover [-f] -m paris protection-group
```

For more information, see [“How to Switch Over Replication From the Primary Cluster to the Secondary Cluster” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

paris resumes its original role as primary cluster for the protection group.

8. **Ensure that the switchover was performed successfully.**

Verify that the protection group is now primary on paris and secondary on newyork and that the state for Data replication and Resource groups is OK on both clusters.

```
# geopg status protection-group
```

Check the runtime status of the application resource group and replication status resource group for the protection group, and the status for each replication status resource associated with a replication component in the protection group.

```
# clresourcegroup status application-rg-in-pg
# clresource status -g application-rg-in-pg
# clresourcegroup status replication-status-rg-for-pg
# clresource status -g replication-status-rg-for-pg
```

Refer to the Status and Status Message fields for the replication status resource of each replication component you want to check.

For more information about the runtime status of replication, see [“Checking the Runtime Status of Oracle Solaris ZFS Snapshot Remote Replication” on page 72](#).

▼ How to Perform a Failback-Takeover on a System That Uses Oracle Solaris ZFS Snapshot Replication

Use this procedure to restart an application on the original primary cluster and use the current data on the original primary cluster. Any updates that occurred on the secondary cluster while it was acting as primary are discarded.

The failback procedures apply only to clusters in a partnership.

Note - To resume using the data on the original primary you must not have replicated data from the current primary to the original primary cluster at any point after the takeover operation on the current primary cluster. To prevent replication between the current primary and the original primary, you must have used the `-n` option whenever you used the `geopg start` command.

Before You Begin Ensure that the following conditions are met:

- If the original primary cluster was down, the cluster is booted and the disaster recovery framework infrastructure is enabled on the cluster. For more information about booting a cluster, see [“Booting a Cluster” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).
- The protection group on the current primary cluster has the `primary` role.
- The protection group on the original primary cluster has either the `primary` role or `secondary` role depending on whether the original primary can be reached during the takeover from the current primary.

This procedure uses the example names `paris` for the original primary cluster and `newyork` for the current primary cluster.

1. On the current primary cluster, `newyork`, stop the protection group locally.

```
newyork-node-1# geopg stop -e local protection-group
```

`-e local`

Specifies the scope of the command. The `local` scope applies the command on the local cluster only.

`protection-group`

Specifies the name of the protection group.

2. Make the protection group primary on `paris` and secondary on `newyork`.

- If `paris` has the `secondary` role, run the following command from `paris`:

```
paris-node-1# geopg takeover protection-group
```

- If `paris` has the `primary` role, run the following command from `newyork`:

```
newyork-node-1# geopg update protection-group
```

3. Validate the configuration of the protection group on each cluster.

Ensure that the protection group is not in an error state. A protection group cannot be started when it is in an error state.

```
paris-node-1# geopg validate protection-group
newyork-node-1# geopg validate protection-group
```

For more information, see [“Validating a Protection Group” in *Installing and Configuring the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

4. From either cluster, start the protection group globally.

```
paris-node-1# geopg start -e global protection-group
```

The protection group on paris now has the primary role, and the protection group on newyork has the role of secondary. The application services are now online on paris.

For more information, see [“How to Activate a Protection Group” in *Administering the Disaster Recovery Framework for Oracle Solaris Cluster 4.4*](#).

5. Ensure that the protection group was started successfully as primary on cluster paris and as secondary on cluster newyork.

Verify that the protection group is now primary on paris and secondary on newyork and that the state for "Data replication" and "Resource groups" is OK on both clusters.

```
# geoadm status
```

Check the runtime status of the application resource group and replication status resource group for the protection group, and the status for each replication status resource associated with a replication component in the protection group.

```
# clresourcegroup status application-rg-in-pg
# clresource status -g application-rg-in-pg
# clresourcegroup status replication-status-rg-for-pg
# clresource status -g replication-status-rg-for-pg
```

Refer to the Status and Status Message fields for the replication status resource of each replication component you want to check.

For more information about the runtime status of replication, see [“Checking the Runtime Status of Oracle Solaris ZFS Snapshot Remote Replication” on page 72](#).

Recovering From an Oracle Solaris ZFS snapshot Remote Replication Error

When an error occurs at the replication level, the error is reflected in the status of the replication status resource of the relevant remote replication component. This corresponding changed status appears in the `Data replication status` field in the relevant protection group's section in the output of the `geoadm status` command.

This section contains the following procedures:

- [“How to Detect Remote Replication Errors” on page 87](#)
- [“How to Recover From an Oracle Solaris ZFS snapshot Remote Replication Error” on page 88](#)

▼ How to Detect Remote Replication Errors

1. Check the status of the replication status resources.

```
paris-node-1# clresource status -v replication-status-rs
```

```
replication-status-rs
```

Specifies the name of the Oracle Solaris ZFS snapshot replication status resource for the replication component

For information about different Resource Status values, see [Table 4, “Status and Status Messages of an Online Oracle Solaris ZFS snapshot Replication Status Resource Group,” on page 74.](#)

Running the `clresource status` command might return output similar to the following example:

```
...
-- Resources --

Resource Name          Node Name          State    Status Message
-----
Resource: replication-status-rs  paris-node-1      Online   Faulted - Failed to get the
list of ZFS snapshots for the ZFS dataset dataset on the host host

Resource: replication-status-rs  paris-node-2      Offline  Offline
...
```

2. Display the aggregate resource status for all components in the protection group.

For example, the output of the `clresource status` command in the preceding example indicates that the replication status resource has reported the `Faulted` status, and therefore the Oracle Solaris ZFS snapshot data replication of the protection group is in the `Error` status on `paris`, as shown below.

```
paris-node-1# geoadm status
Cluster: paris

Partnership "paris-newyork-ps" : OK
Partner clusters      : newyork
Synchronization      : OK
ICRM Connection      : OK

Heartbeat "paris-newyork-ps" monitoring "newyork": OK
Heartbeat plug-in "ping_plugin"      : Inactive
Heartbeat plug-in "tcp_udp_plugin"   : OK

Protection group "zfssnap-pg" : Error
Partnership      : paris-newyork-ps
Synchronization : OK

Cluster paris    : Error
Role             : Primary
PG activation state : Activated
Configuration    : OK
Data replication : Error
Resource groups  : OK

Cluster newyork : Error
Role            : Secondary
PG activation state : Activated
Configuration    : OK
Data replication : Error
Resource groups  : OK
```

▼ How to Recover From an Oracle Solaris ZFS snapshot Remote Replication Error

To recover from an error state, you might perform some or all of the steps in the following procedure.

1. **Check the status of the replication status resource to determine whether the data replication has experienced an error. The status message of the resource indicates why the resource reported the `Faulted` status. The status values table shown in Table 4, “Status and Status Messages of an Online Oracle Solaris ZFS snapshot Replication Status Resource Group,” on page 74 will also help to understand the error. Fix the errors according to the reason cited in the message.**

If required, check the resource log, console and syslog messages, and trace log for more information about the errors. For more information, see “[Debugging an Oracle Solaris ZFS Snapshot Protection Group](#)” on page 40.

2. **Revalidate the protection group configuration.**

```
paris-node-1# geopg validate protection-group
```

- If the `geopg validate` command determines that the configuration is valid, the state of the protection group changes to reflect that fact.
- If the configuration is not valid, the `geopg validate` command will fix any internal replication infrastructure issues like resource group affinities and resource properties that are automatically managed by the replication setup software. In case of any failures in such attempt, or if there are issues with the replication infrastructure prerequisites that should be fixed manually, the command returns a failure message citing the problem.

3. **Review the protection group configuration.**

```
paris-node-1# geopg show protection-group
```

4. **Review the configuration status and runtime status of the protection group.**

```
paris-node-1# geoadm status
```


Disaster Recovery Framework Properties for Oracle Solaris ZFS Snapshot

This appendix provides the properties of the replication resources related to the Oracle Solaris ZFS Snapshot replication module in disaster recovery framework.

This appendix contains the following sections:

- [“Extension Properties for Oracle Solaris ZFS Snapshot Replication Agent Resource” on page 91](#)
- [“Extension Properties for Oracle Solaris ZFS Snapshot Resource Status” on page 95](#)

The disaster recovery framework software automatically creates resource of the ORCL.ZFSSnapRepAgent and ORCL.ZFSSnapRepStatus resource types for each configured ZFS snapshot replication component, and internally manages the properties for such resources.

Extension Properties for Oracle Solaris ZFS Snapshot Replication Agent Resource

Note - You must not edit the extension properties of a resource of type ORCL.ZFSSnapRepAgent by using the Oracle Solaris Cluster commands. You can edit the values of the properties by modifying the configuration file. For more information about the configuration file, see [“Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File” on page 26](#).

The following list describes the extension properties of a resource of type ORCL.ZFSSnapRepAgent that you must not edit when the resource is configured in a ZFS snapshot protection group:

PG

Specifies the protection group to which this resource is associated.

Data Type: String

Category: Required

Tuning Recommendations: This property can be set only at creation

Default: None

Role

Specifies the replication role of this resource

Data Type: Enum

Possible values: PRIMARY and SECONDARY

Category: Optional

Tuning Recommendations: This property can be tuned at any time

Default: SECONDARY

Local_rep_user

Specifies the username that is used for replication on local cluster

Data Type: String

Category: Required

Tuning Recommendations: This property is tuned when disabled

Default: None

Remote_rep_user

Specifies the username that is used for replication on remote cluster

Data Type: String

Category: Required

Tuning recommendations: This property is tuned when disabled

Default: None

Passphrase_pstring_name

Specifies the name of the Oracle Solaris Cluster private string object holding the SSH passphrase of the local cluster replication user

Data Type: String

Category: Required

Tuning recommendations: This property is tuned when disabled

Default: None

Private_key_file

Specifies the path to the SSH private key file for the local cluster replication user

Data Type: String

Category: Optional

Tuning recommendations: This property is tuned when disabled

Default: Default value that is used by the ssh-add utility

Remote_host

Specifies the name of the logical host on the remote side for replication

Data Type: String

Category: Required

Tuning recommendations: This property is tuned when disabled

Default: None

Remote_zone

Specifies the zone name of the remote partner cluster

Data Type: String

Category: Optional

Tuning recommendations: This property is tuned when disabled

Default: global

Local_dataset

Defines the local ZFS dataset that is used as replication source

Data Type: String

Category: Required

Tuning recommendations: This property is tuned when disabled

Default: None

Remote_dataset

Defines the remote ZFS dataset that is used as replication target

Data Type: String

Category: Optional

Tuning recommendations: This property is tuned when disabled

Default: Value of the Local_dataset

Replication_interval

Specifies interval in seconds between ZFS snapshot replication cycles

Data Type: Integer

Category: Optional

Tuning recommendations: This property can be tuned any time

Default: 120

Minimum: 0

Num_of_snapshots_to_store

Specifies the number of successfully replicated snapshots that must be stored on local and remote side

Data Type: Integer

Category: Optional

Tuning recommendations: This property can be tuned any time

Default: 2

Minimum: 1

Replication_stream_package

Specifies whether the replication must use Replication Stream Package

Data Type: Boolean

Category: Optional

Tuning recommendations: This property must be tuned when disabled

Default: FALSE

Send_properties

Specifies whether ZFS dataset properties must be part of the snapshot streams

Data Type: Boolean

Category: Optional

Tuning recommendations: This property must be tuned when disabled

Default: TRUE

Intermediary_snapshots

Specifies whether intermediary snapshots between replication snapshots must be sent and received

Data Type: Boolean

Category: Optional

Tuning recommendations: This property can be tuned any time

Default: FALSE

Recursive

Specifies whether to replicate the entire hierarchy of the local dataset, that is, the local dataset and its children

Data Type: Boolean

Category: Optional

Tuning recommendations: This property must be tuned when disabled

Default: TRUE

Note - Only one parameter out of Recursive and Intermediary_snapshots can be TRUE at the same time. Only one parameter out of Recursive and Replication_stream_package can be TRUE at the same time.

Extension Properties for Oracle Solaris ZFS Snapshot Resource Status

Note - You must not edit the extension properties of a resource of type ORCL.ZFSSnapRepStatus using the Oracle Solaris Cluster commands. You can edit the values of the properties by modifying the configuration file. For more information about the configuration file, see [“Overview of the Oracle Solaris ZFS Snapshot Remote Replication Configuration File”](#) on page 26.

The following list describes the extension properties of a resource of type ORCL.ZFSSnapRepStatus:

PG	<p>Specifies the protection group to which this resource is associated.</p> <p>Data Type: String</p> <p>Category: Required</p> <p>Tuning Recommendations: This property can be set only at creation</p> <p>Default: None</p>
Role	<p>Specifies the replication role of this resource</p> <p>Data Type: Enum</p>

Possible values: PRIMARY and SECONDARY

Category: Optional

Tuning Recommendations: This property can be tuned at any time

Default: SECONDARY

Replication_agent_rs Specifies the name of the replication agent resource corresponding to the replication component

Data Type: String

Category: Required

Tuning recommendations: This property must be tuned when disabled

Default: None

Local_dataset Specifies the local ZFS dataset that is configured in the replication component

Data Type: String

Category: Required

Tuning recommendations: This property must be tuned when disabled

Default: None

Remote_dataset Specifies Remote ZFS dataset configured in replication component

Data Type: String

Category: Optional

Tuning recommendations: This property must be tuned when disabled

Default: Value of the Local_dataset

Local_zpool_rs Specifies the name of the local cluster's infrastructure HAStoragePlus resource managing the zpool for the replication component

Data Type: String

Category: Required

Tuning recommendations: This property must be tuned when disabled

Default: None

Local_rep_user Specifies the username used for replication on local cluster

Data Type: String

Category: Required

Tuning recommendations: This property must be tuned when disabled

	Default: None
Remote_rep_user	<p>Specifies username used for replication on remote cluster</p> <p>Data Type: String</p> <p>Category: Required</p> <p>Tuning recommendations: This property must be tuned when disabled</p> <p>Default: None</p>
Remote_host	<p>Specifies the name of the logical host on the remote side for replication</p> <p>Data Type: String</p> <p>Category: Required</p> <p>Tuning recommendations: This property must be tuned when disabled</p> <p>Default: None</p>
Passphrase_pstring_name	<p>Specifies the name of the Oracle Solaris Cluster private string object holding the SSH passphrase of the local cluster replication user</p> <p>Data Type: String</p> <p>Category: Required</p> <p>Tuning recommendations: This property must be tuned when disabled</p> <p>Default: None</p>
Private_key_file	<p>Specifies the path to the SSH private key file for the local cluster replication user</p> <p>Data Type: String</p> <p>Category: Optional</p> <p>Tuning recommendations: This property must be tuned when disabled</p> <p>Default: Default value used by the ssh-add utility</p>
Remote_zone	<p>Specifies the zone name of the remote partner cluster</p> <p>Data Type: String</p> <p>Category: Optional</p> <p>Tuning recommendations: This property must be tuned when disabled</p> <p>Default: global</p>

Use Cases for Oracle Solaris ZFS Snapshot Replication

This appendix provides example use cases for configuring the Oracle Solaris ZFS Snapshots replication in disaster recovery framework.

This appendix contains the following sections:

- [“Configuration Assumptions of Use Cases” on page 99](#)
- [“Use Case: Setting Up Replication User and SSH” on page 100](#)
- [“Use Case: Configuring Oracle Solaris ZFS Snapshot Replication When Both Partner Clusters Are Global Zones” on page 108](#)
- [“Use Case: Configuring Oracle Solaris ZFS Snapshot Replication With One Partner Cluster as Zone Cluster” on page 113](#)
- [“Use Case: Configuring Oracle Solaris ZFS Snapshot Replication With Both Partner Clusters as Zone Cluster” on page 119](#)
- [“Use Case: Configuring Oracle Solaris ZFS Snapshot Replication With Zpools for Globally Mounted ZFS Filesystems” on page 125](#)

Configuration Assumptions of Use Cases

The Oracle Solaris ZFS snapshot replication is set up using the `zfs_snap_geo_register` setup script. This examples uses a setup consisting of clusters `paris` and `newyork` that are in a partnership.

The following configuration must exist in the cluster before running the setup script:

- Clusters `paris` and `newyork` are in a partnership known as `paris-newyork`. Primary cluster `paris` has two nodes: `paris-node-1` and `paris-node-2`. Secondary cluster `newyork` has two nodes: `newyork-node-1` and `newyork-node-2`.
- The application resource group `app1-rg1` is configured on both the partner clusters.
- On each cluster, the `app1-rg1` resource group uses a `HAStoragePlus` resource to manage the ZFS storage pool that the application uses. In the primary cluster, the application uses

the HAStoragePlus resource `par-app1-hasp1` to manage the ZFS storage pool `srcpool1`. Similarly, the HAStoragePlus resource `ny-app1-hasp1` is used on the secondary cluster to manage the ZFS storage pool `targpool1`.

- The application `app1-rg1` on the primary cluster `paris` uses the ZFS dataset `srcpool1/app1-ds1`, and the application `app1-rg1` on the secondary cluster `newyork` uses the ZFS dataset `targpool1/app1-ds1-copy`.
- On both clusters, the application uses the same application user `appuser1` to create and access application data stored in the ZFS datasets that the application uses.

Use Case: Setting Up Replication User and SSH

The Oracle Solaris ZFS snapshot replication uses SSH authentication to transfer ZFS snapshot streams on to the remote replication target host and to execute commands on the remote host. The replication user on each cluster node must be able to SSH as remote cluster's replication user to each remote cluster node. Therefore, you must set up SSH authentication for the replication users to enable communication between the local and remote cluster. You must use the remote logical hostname as the SSH target.

You must set up the SSH authentication before creating the protection group for Oracle Solaris ZFS snapshot replication. This example uses Unix users and groups. The Unix user `zfsuser1` is the replication user on the `paris` cluster. The Unix user `zfsuser2` is the replication user on the `newyork` cluster. This example describes how to set up SSH configuration between the two replication users on the partner clusters. The host keys are stored in `.ssh/known_hosts` file of the user's home directory. The SSH keys are generated and stored in `.ssh` directory in the user's home directory. The authorized public keys are stored in the `.ssh/authorized_keys` file in the home directory. This example uses the default SSH private key file on `paris`, and a non-default file on `newyork`.

The replication user and the SSH setup are all created in the global zones of the partner clusters, even if a partner cluster is a zone cluster. All the commands shown below must be executed in the global zone of the partner clusters, as mentioned in the steps.

This procedure describes the steps to set up SSH for replication between `paris` and `newyork` partner clusters. All the commands must be executed in the global zone of the partner clusters.

1. Assume the root role in the global zone on both the nodes in the primary cluster `paris`.
2. Type the following commands in the global zone of both the nodes in the primary cluster `paris` to create the replication user `zfsuser1`.

```
phys-paris # groupadd -g 101 user1grp
phys-paris # useradd -u 1234 -g 101 -d /export/home/zfsuser1 zfsuser1
phys-paris # mkdir /export/home/zfsuser1
```

```
phys-paris # chown zfsuser1 /export/home/zfsuser1
phys-paris # chgrp user1grp /export/home/zfsuser1
phys-paris # passwd zfsuser1
```

3. Log in as the replication user zfsuser1 in the global cluster nodes of both the nodes of the primary cluster.
4. Create SSH keys in the global cluster nodes of both the nodes in the primary cluster paris.

```
phys-paris $ mkdir ~/.ssh
phys-paris $ ssh-keygen -t rsa
```

Assume that the default file used by ssh-keygen to store the key is ~/.ssh/id_rsa. Ensure that you specify the same passphrase to protect the private key in the global cluster nodes of both the nodes.

5. Assume the root role in the global cluster nodes of both the nodes in the secondary cluster newyork.
6. Create the replication user zfsuser2 in the global cluster nodes of both the nodes in the secondary cluster newyork.

```
phys-newyork # groupadd -g 102 user2grp
phys-newyork # useradd -u 5678 -g 102 -d /export/home/zfsuser2 zfsuser2
phys-newyork # mkdir /export/home/zfsuser2
phys-newyork # chown zfsuser2 /export/home/zfsuser2
phys-newyork # chgrp user2grp /export/home/zfsuser2
phys-newyork # passwd zfsuser2
```

7. Log in as the replication user zfsuser2 in the global cluster nodes of both the nodes of the secondary cluster.
8. Create SSH keys in a non-default file ~/.ssh/zfsrep1 in the global cluster nodes of both the nodes in the secondary cluster newyork.

```
phys-newyork $ mkdir ~/.ssh
phys-newyork $ ssh-keygen -t rsa -f ~/.ssh/zfsrep1
```

Ensure that you specify the same passphrase to protect the private key in the global cluster nodes of both the nodes .

9. Copy SSH public key of replication user from the global cluster node of each node of the primary cluster to the global cluster node of each node of the secondary cluster and vice versa. Use the passwords of the users to perform authentication when using scp.
 - Copy the SSH public key from the global cluster node of the primary cluster node paris-node-1 to the global cluster node of the secondary cluster node newyork-node-1. Type the following command as zfsuser1 in the global zone of paris-node-1:

```
phys-paris-1 $ scp ~/.ssh/id_rsa.pub zfsuser2@phys-newyork-1:~/.ssh/paris-node-1
```

Type the following commands as `zfsuser2` in the global cluster node of `newyork-node-1`:

```
phys-newyork-1 $ cat ~/.ssh/paris-node-1 >>~/.ssh/authorized_keys
phys-newyork-1 $ rm ~/.ssh/paris-node-1
```

- Copy the SSH public key from the global cluster node of the primary cluster node `paris-node-2` to the global cluster node of the secondary cluster node `newyork-node-1`. Type the following commands as `zfsuser1` in the global cluster node of `paris-node-2`:

```
phys-paris-2 $ scp ~/.ssh/id_rsa.pub zfsuser2@phys-newyork-1:~/.ssh/paris-node-2
```

Type the following commands as `zfsuser2` in the global cluster node of `newyork-node-1`:

```
phys-newyork-1 $ cat ~/.ssh/paris-node-2 >>~/.ssh/authorized_keys
phys-newyork-1 $ rm ~/.ssh/paris-node-2
```

- Copy the SSH public key from the global cluster node of the primary cluster node `paris-node-1` to the global cluster node of the secondary cluster node `newyork-node-2`. Type the following command as `zfsuser1` in the global cluster node of `paris-node-1`:

```
phys-paris-1 $ scp ~/.ssh/id_rsa.pub zfsuser2@phys-newyork-2:~/.ssh/paris-node-1
```

Type the following commands as `zfsuser2` in the global cluster node of `newyork-node-2`:

```
phys-newyork-2 $ cat ~/.ssh/paris-node-1 >>~/.ssh/authorized_keys
phys-newyork-2 $ rm ~/.ssh/paris-node-1
```

- Copy the SSH public key from the global cluster node of the primary cluster node `paris-node-2` to the global cluster node of the secondary cluster node `newyork-node-2`. Type the following commands as `zfsuser1` in the global cluster node of `paris-node-2`:

```
phys-paris-2 $ scp ~/.ssh/id_rsa.pub zfsuser2@phys-newyork-2:~/.ssh/paris-2
```

Type the following commands as `zfsuser2` in the global cluster node of `newyork-node-2`:

```
phys-newyork-2 $ cat ~/.ssh/paris-node-2 >>~/.ssh/authorized_keys
phys-newyork-2 $ rm ~/.ssh/paris-node-2
```

- Copy the SSH public key from the global cluster node of the secondary cluster node `newyork-node-1` to the global cluster node of the primary cluster node `paris-node-1`. Type the following commands as `zfsuser2` in the global cluster node of `newyork-node-1`:

```
phys-newyork-1 $ scp ~/.ssh/zfsrep1.pub
zfsuser1@phys-paris-1:~/.ssh/phys-newyork-1
```

Type the following command as zfsuser1 in the global cluster node of paris-node-1:

```
phys-paris-1 $ cat ~/.ssh/newyork-node-1 >>~/.ssh/authorized_keys
phys-paris-1 $ rm ~/.ssh/newyork-node-1
```

- Copy the SSH public key from the global cluster node of the secondary cluster node newyork-node-2 to the global cluster node of the primary cluster node paris-node-1. Type the following commands as zfsuser2 in the global cluster node of newyork-node-2:

```
phys-newyork-2 $ scp ~/.ssh/zfsrep1.pub
zfsuser1@phys-paris-1:~/.ssh/phys-newyork-2
```

Type the following commands as zfsuser1 in the global cluster node of paris-node-1:

```
phys-paris-1 $ cat ~/.ssh/newyork-node-2 >>~/.ssh/authorized_keys
phys-paris-1 $ rm ~/.ssh/newyork-node-2
```

- Copy the SSH public key from the global cluster node of the secondary cluster node newyork-node-1 to the global cluster node of the primary cluster node paris-node-2. Type the following commands as zfsuser2 in the global cluster node of newyork-node-1:

```
phys-newyork-1 $ scp ~/.ssh/zfsrep1.pub
zfsuser1@phys-paris-2:~/.ssh/phys-newyork-1
```

Type the following command as zfsuser1 in the global cluster node of paris-node-2:

```
phys-paris-2 $ cat ~/.ssh/newyork-node-1 >>~/.ssh/authorized_keys
phys-paris-2 $ rm ~/.ssh/newyork-node-1
```

- Copy the SSH public key from the global cluster node of the secondary cluster node newyork-node-2 to the global cluster node of the primary cluster node paris-node-2. Type the following commands as zfsuser2 in the global cluster node of newyork-node-2:

```
phys-newyork-2 $ scp ~/.ssh/zfsrep1.pub
zfsuser1@phys-paris-2:~/.ssh/phys-newyork-2
```

Type the following commands as zfsuser1 in the global cluster node of paris-node-2:

```
phys-paris-2 $ cat ~/.ssh/newyork-node-2 >>~/.ssh/authorized_keys
phys-paris-2 $ rm ~/.ssh/newyork-node-2
```

10. Add the entries for `paris-lh` in the `/etc/hosts` file in the global cluster node on all nodes of the primary and secondary cluster. Similarly, add the entries for `newyork-lh` in the `/etc/hosts` file in the global cluster node on all nodes of the primary and secondary cluster.
11. Each replication user's `.ssh/known_hosts` requires multiple entries for remote cluster's replication logical hostname, which means that there is one logical hostname entry corresponding to the host key of each remote cluster node. Since the SSH will use logical hostname as target, so it is required to add the entries for remote cluster's logical hostname on to each local cluster's node. Assume that the IP address for the logical hostname `paris-lh` is `172.16.2.75`. Assume that the IP address for the logical hostname `newyork-lh` is `192.168.15.125`. Perform the following steps to add logical host key entries to `known_hosts` SSH file

- a. Login to the global cluster node on each node on `paris` as the `zfsuser1` user and edit the `~/.ssh/known_hosts` file in the home directory. Duplicate each entry relevant to global cluster node of `newyork` cluster nodes, and change each such duplicate entry with `newyork-lh,192.168.15.125`.

The `known_hosts` file for the replication user `zfsuser1` on each node `paris-node-M` must contain the following entries :

```
phys-paris $ cat ~/.ssh/known_hosts
phys-newyork-1,phys-newyork-1-IP ssh-rsa big key for phys-newyork-1
newyork-lh,192.168.15.125 ssh-rsa same big key for phys-newyork-1
phys-newyork-2,phys-newyork-2-IP ssh-rsa big key for phys-newyork-2
newyork-lh,192.168.15.125 ssh-rsa same big key for phys-newyork-2
```

- b. Login to the global cluster node on each node on `newyork` as the `zfsuser2` user and edit the `.ssh/known_hosts` file in the home directory. Duplicate each entry relevant to global cluster node of `paris` cluster nodes and change each of the duplicate entries for `paris` with `paris-lh,172.16.2.75`.

```
phys-newyork $ cat ~/.ssh/known_hosts
phys-paris-1,phys-paris-1-IP ssh-rsa big key for global zone of phys-paris-1
paris-lh,172.16.2.75 ssh-rsa same big key for global zone of phys-paris1
phys-paris-2,phys-paris-2-IP ssh-rsa big key for phys-paris-2
paris-lh,172.16.2.75 ssh-rsa same big key for phys-paris-2
```

The replication feature requires that the entries for replication logical hostname are available in the `known_hosts` file used by SSH. Replication does not require these entries for cluster node names in such file. Hence, you can remove these entries after above actions.

12. Verify whether the replication user in the global cluster node on each cluster node of one partner cluster can connect using SSH to the replication-related logical hostname in the global cluster node on each node of the other partner cluster as the remote cluster's

replication user. You can perform the following actions to verify that the SSH connection from global cluster node of paris to global cluster node of newyork is working correctly:

Perform the following steps to check the SSH connection from paris to newyork.

- a. Create a logical hostname resource group and resource to host newyork-lh in the global cluster node of the secondary cluster newyork and bring it online. Type the following commands from the global cluster node of any one node of newyork cluster.

```
phys-newyork-1 # /usr/cluster/bin/clrt register SUNW.LogicalHostname
phys-newyork-1 # /usr/cluster/bin/clrg create repl-lh-rg
phys-newyork-1 # /usr/cluster/bin/clrslh create -g repl-lh-rg \
-h newyork-lh repl-lh-rs
phys-newyork-1 # /usr/cluster/bin/clrg online -eM repl-lh-rg
```

- b. Verify whether the SSH connection works from the global cluster node of each node of paris cluster to the logical hostname hosted on the global cluster node of newyork-node-1.

- i. Type the following command in the global cluster node of any one node of the newyork cluster to switch the logical hostname to the global cluster node of newyork-node-1.

```
phys-newyork-1 # clrg switch -n phys-newyork-1 repl-lh-rg
```

- ii. Type the following command to check that zfsuser1 in the global cluster node of paris-node-1 can SSH as zfsuser2 to the global cluster node of newyork-node-1 where newyork-lh is hosted currently.

```
phys-paris-1 $ ssh zfsuser2@newyork-lh
```

When prompted, enter the passphrase to connect to the remote system.

- iii. Type the following command to check that zfsuser1 in the global cluster node of paris-node-2 can SSH as zfsuser2 to the global cluster node of newyork-node-1 where newyork-lh is hosted currently.

```
phys-paris-2 $ ssh zfsuser2@newyork-lh
```

When prompted, enter the passphrase to connect to the remote system.

- c. Verify whether the SSH connection works from the global cluster node of each node of paris cluster to the logical hostname hosted on the global cluster node of newyork-node-2.

- i. Type the following command in the global cluster node of any one node of the newyork cluster to switch the logical hostname to the global cluster node of newyork-node-2.

```
phys-newyork-1 # clrg switch -n phys-newyork-2 repl-lh-rg
```

- ii. Type the following command to check that zfsuser1 in the global cluster node of paris-node-1 can SSH as zfsuser2 to the global cluster node of newyork-node-2 where newyork-lh is hosted currently.

```
phys-paris-1 $ ssh zfsuser2@newyork-lh
```

When prompted, enter the passphrase to connect to the remote system.

- iii. Type the following command to check that zfsuser1 in the global cluster node of paris-node-2 can SSH as zfsuser2 to the global cluster node of newyork-node-2 where newyork-lh is hosted currently.

```
phys-paris-2 $ ssh zfsuser2@newyork-lh
```

When prompted, enter the passphrase to connect to the remote system.

- d. Remove the logical hostname test setup from the secondary cluster. Type the following commands from the global cluster node of any one node of newyork cluster.

```
phys-newyork-1 # /usr/cluster/bin/clrs delete -F repl-lh-rs
```

Perform the following steps to check the SSH connection from newyork to paris.

- a. Create a logical hostname resource group and resource to host paris-lh in the global cluster node of the primary cluster paris and bring it online. Type the following commands from the global cluster node of any one node of the paris cluster.

```
phys-paris-1 # /usr/cluster/bin/clrt register SUNW.LogicalHostname
```

```
phys-paris-1 # /usr/cluster/bin/clrg create repl-lh-rg
```

```
phys-paris-1 # /usr/cluster/bin/clrslh create -g repl-lh-rg \  
-h paris-lh repl-lh-rs
```

```
phys-paris-1 # /usr/cluster/bin/clrg online -eM repl-lh-rg
```

- b. Verify whether the SSH connection works from the global cluster node of each node of newyork cluster to the logical hostname hosted on the global cluster node of paris-node-1.

- i. Type the following command in the global cluster node of any one node of the paris cluster to switch the logical hostname to the global cluster node of paris-node-1.

```
phys-paris-1 # clrg switch -n phys-paris-1 repl-lh-rg
```

- ii. Type the following command to check that `zfsuser2` in the global cluster node of `newyork-node-1` can SSH as `zfsuser1` to the global cluster node of `paris-node-1` where `paris-lh` is hosted currently.

```
phys-newyork-1 $ ssh zfsuser1@paris-lh -i ~/.ssh/zfsrep1
```

When prompted, enter the passphrase to connect to the remote system.

- iii. Type the following command to check that `zfsuser2` in the global cluster node of `newyork-node-2` can SSH as `zfsuser1` to the global cluster node of `paris-node-1` where `paris-lh` is hosted currently.

```
phys-newyork-2 $ ssh zfsuser1@paris-lh -i ~/.ssh/zfsrep1
```

When prompted, enter the passphrase to connect to the remote system.

- c. Verify whether the SSH connection works from the global cluster node of each node of `newyork` cluster to the logical hostname hosted on the global cluster node of `paris-node-2`.
 - i. Type the following command in the global cluster node of any one node of the `paris` cluster to switch the logical hostname to the global cluster node of `paris-node-2`.

```
phys-paris-1 # clrg switch -n phys-paris-2 repl-lh-rg
```

- ii. Type the following command to check that `zfsuser2` in the global cluster node of `newyork-node-1` can SSH as `zfsuser1` to the global cluster node of `paris-node-2` where `paris-lh` is hosted currently.

```
phys-newyork-1 $ ssh zfsuser1@paris-lh -i ~/.ssh/zfsrep1
```

When prompted, enter the passphrase to connect to the remote system.

- iii. Type the following command to check that `zfsuser2` in the global cluster node of `newyork-node-2` can SSH as `zfsuser1` to the global cluster node of `paris-node-2` where `paris-lh` is hosted currently.

```
phys-newyork-2 $ ssh zfsuser1@paris-lh -i ~/.ssh/zfsrep1
```

When prompted, enter the passphrase to connect to the remote system.

- d. Remove the logical hostname test setup from the primary cluster. Type the following commands from the global cluster node of any one node of `paris` cluster.

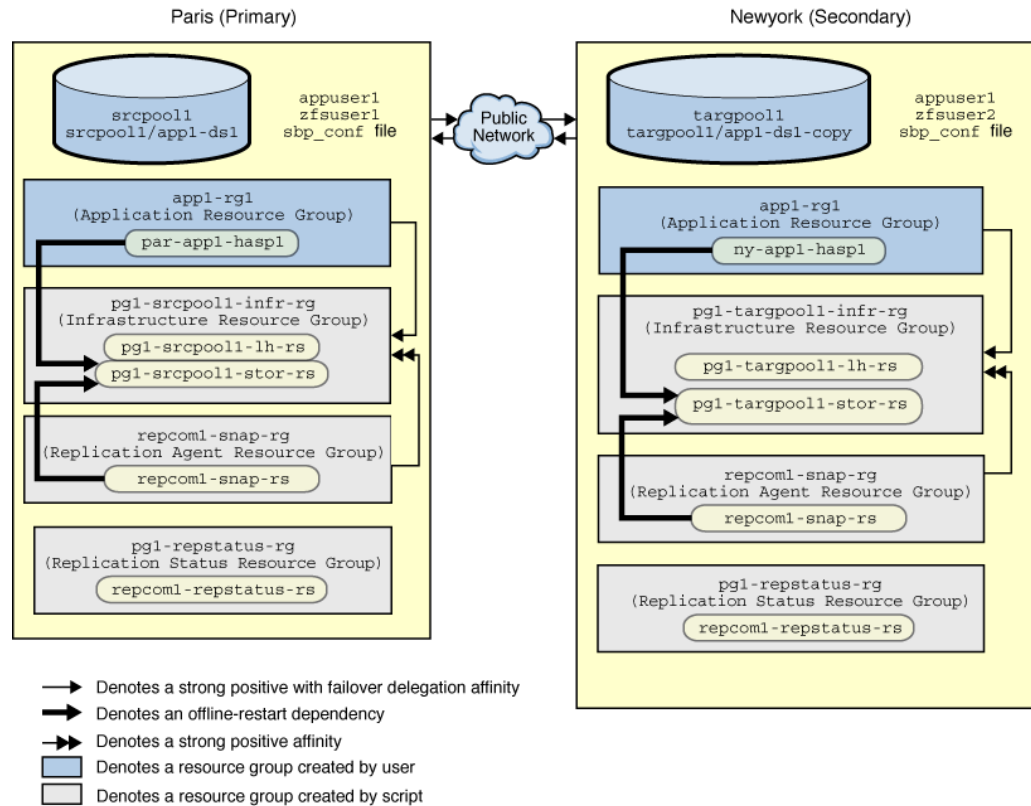
```
phys-paris-1 # /usr/cluster/bin/clrs delete -F repl-lh-rs
```

Use Case: Configuring Oracle Solaris ZFS Snapshot Replication When Both Partner Clusters Are Global Zones

This example shows how to set up the protection group with Oracle Solaris ZFS snapshot replication to protect and manage the application and its ZFS datasets. Assume that the application resource group and application user is already set up. For more information about the configuration assumptions for this example, see [“Configuration Assumptions of Use Cases” on page 99](#).

The following figure shows the resource groups and resources that are created by setup actions performed by the `zfs_snap_geo_register` script.

FIGURE 2 Example Setup of Oracle Solaris ZFS Snapshot Replication in the Global Zone on Both the Partner Clusters



This figure displays the infrastructure resource group, replication agent resource group, and replication status resource group apart from the application resource group. The infrastructure resource group contains the logical hostname resource and the HAStoragePlus resource. The replication agent resource group contains a replication agent resource. The replication status resource group contain the replication status resource.

The setup actions done by the `zfs_snap_geo_register` script also sets the following extension properties on the resources:

- The `ZpoolImportOnly` property is set to `True` on HAStoragePlus resource in the infrastructure resource group. This ensure that when the resource starts, it imports the ZFS storage pool without mounting the file systems.

Note - The `ZpoolsImportOnly` property of `SUNW.HAStoragePlus` resource type is used by the cluster software internally for the replication infrastructure.

- The `ZpoolsExportOnStop` property is set to `False` on `HAStoragePlus` resource in the application resource group. This ensures that when the resource stops, it does not export the ZFS storage pool. This is required as the infrastructure `HAStoragePlus` resource manages the export of the ZFS storage pool after both application and replication using the ZFS storage pool stop.

Note - The `ZpoolsExportOnStop` property of `SUNW.HAStoragePlus` resource type is used by the cluster software internally for the replication infrastructure.

You can perform the following steps to configure the Oracle Solaris ZFS snapshot replication by using the `zfs_snap_geo_register` script:

1. Decide the hostname to use on each cluster as replication-related logical hostname. For this example, assume that the logical hostname on the `paris` cluster is `paris-lh`. The logical hostname on the `newyork` cluster is `newyork-lh`.

Add the logical hostname entries for `paris-lh` in the `/etc/hosts` file on both the nodes of the `paris` cluster. Similarly, add the logical hostname entries for `newyork-lh` in the `/etc/hosts` file on both the nodes of the `newyork` cluster.

2. Create the replication user and SSH setup. For information about the step for creating the replication user and SSH setup, see [“Use Case: Setting Up Replication User and SSH” on page 100](#).
3. Provide ZFS permissions to replication user on both clusters. You must import the ZFS storage pool to provide permissions. If application's `HAStoragePlus` is online, log in as root user on the node where it is online. Else if the zpool is not online on any cluster node, import the zpool on any node and perform ZFS commands on that node. On each cluster, log in as the root user to the cluster node where the zpool is imported.

Type the following commands on the node of the `paris` cluster where `srcpool1` is imported:

```
# /sbin/zfs allow zfsuser1
  create,destroy,hold,mount,receive,release,send,rollback,snapshot \
srcpool1/app1-ds1
# /sbin/zfs allow srcpool1/app1-ds1
---- Permissions on srcpool1/app1-ds1 -----
Local+Descendent permissions:
user zfsuser1 create,destroy,hold,mount,receive,release,rollback,send,snapshot
```

#

Type the following commands on the node of the newyork cluster where targpool1 is imported:

```
# /sbin/zfs allow zfsuser2
  create,destroy,hold,mount,receive,release,send,rollback,snapshot \
targpool1/app1-ds1-copy
# /sbin/zfs allow targpool1/app1-ds1-copy
---- Permissions on targpool1/app1-ds1-copy -----
Local+Descendent permissions:
user zfsuser2 create,destroy,hold,mount,receive,release,rollback,send,snapshot
#
```

Note that the replication user on each cluster must have the above Local+Descendent ZFS permissions on the ZFS dataset used on that cluster.

- Suppose you create a file `/var/tmp/geo/zfs_snapshot/sbp_conf` file to use as the script-based plugin configuration file on both clusters to specify replication component node lists and evaluation rules. Add the following entry in the `/var/tmp/geo/zfs_snapshot/sbp_conf` file on each node of the paris cluster :

```
repcom1|any|paris-node-1,paris-node-2
```

Add the following entry in the `/var/tmp/geo/zfs_snapshot/sbp_conf` file on each node of the newyork cluster :

```
repcom1|any|newyork-node-1,newyork-node-2
```

- Copy `/opt/ORCLscgprepzfssnap/etc/zfs_snap_geo_config` to create a parameters file `/var/tmp/geo/zfs_snapshot/repcom1_conf` for the replication component. Type the configuration parameters in the configuration file.

```
PS=paris-newyork
PG=pg1
REPCOMP=repcom1
REPRS=repcom1-repstatus-rs
REPRG=pg1-repstatus-rg
DESC="Protect app1-rg1 using ZFS snapshot replication"
APPRG=app1-rg1
CONFIGFILE=/var/tmp/geo/zfs_snapshot/sbp_conf
LOCAL_REP_USER=zfsuser1
REMOTE_REP_USER=zfsuser2
LOCAL_PRIV_KEY_FILE=
REMOTE_PRIV_KEY_FILE=/export/home/zfsuser2/.ssh/zfsrep1
LOCAL_ZPOOL_RS=par-app1-haspl
```

```

REMOTE_ZPOOL_RS=ny-app1-haspl
LOCAL_LH=paris-lh
REMOTE_LH=newyork-lh
LOCAL_DATASET=srcpool1/app1-ds1
REMOTE_DATASET=targpool1/app1-ds1-copy
REPLICATION_INTERVAL=120
NUM_OF_SNAPSHOTS_TO_STORE=2
REPLICATION_STREAM_PACKAGE=false
SEND_PROPERTIES=true
INTERMEDIARY_SNAPSHOTS=false
RECURSIVE=true
MODIFY_PASSPHRASE=false

```

6. Execute the `zfs_snap_geo_register` script using the replication configuration file as parameter from any one node of paris cluster. specify the SSH passphrase for `zfsuser1` of paris cluster at the Password for property `local_service_password` prompt and the SSH passphrase for `zfsuser2` of newyork cluster at the Password for property `remote_service_password` prompt

```

# /opt/ORCLscgprepzfssnap/util/zfs_snap_geo_register -f /var/tmp/geo/zfs_snapshot/
repcom1_conf
Password for property local_service_password :
Password for property remote_service_password :

```

This setup action performed by the `zfs_snap_geo_register` script creates the following components in the primary cluster, as shown in [Figure 2, “Example Setup of Oracle Solaris ZFS Snapshot Replication in the Global Zone on Both the Partner Clusters,”](#) on page 109:

- Protection group `pg1`
 - Replication component `repcom1`
 - Infrastructure resource group `pg1-srcpool1-infr-rg` and its resources as shown in the figure
 - Replication resource group `repcom1-snap-rg` which contains the resource `repcom1-snap-rs`
 - Replication status resource group `pg1-repstatus-rg` and replication status resource `repcom1-repstatus-rs`
7. On any node of the paris cluster, check that the protection group and replication component is created successfully.

```

# /usr/cluster/bin/geopg show pg1
# /usr/cluster/bin/geopg status pg1

```


8. On any node of the `newyork` cluster, get the protection group that is created on the primary cluster.

```
# /usr/cluster/bin/geopg get -s paris-newyork pg1
```

This command creates the resource groups and resources in the secondary cluster, as shown in [Figure 2, “Example Setup of Oracle Solaris ZFS Snapshot Replication in the Global Zone on Both the Partner Clusters,”](#) on page 109.

9. From any node of the `newyork` cluster, check that the protection group and replication components are available. Ensure that the protection group synchronization status between `paris` and `newyork` cluster shows OK.

```
# /usr/cluster/bin/geopg show pg1
# /usr/cluster/bin/geopg status pg1
```

Similarly, check status from any node of `paris` cluster:

```
# /usr/cluster/bin/geopg show pg1
# /usr/cluster/bin/geopg status pg1
```

10. Activate the protection group to start the Oracle Solaris ZFS snapshot replication.

```
$ /usr/cluster/bin/geopg start -e global pg1
```

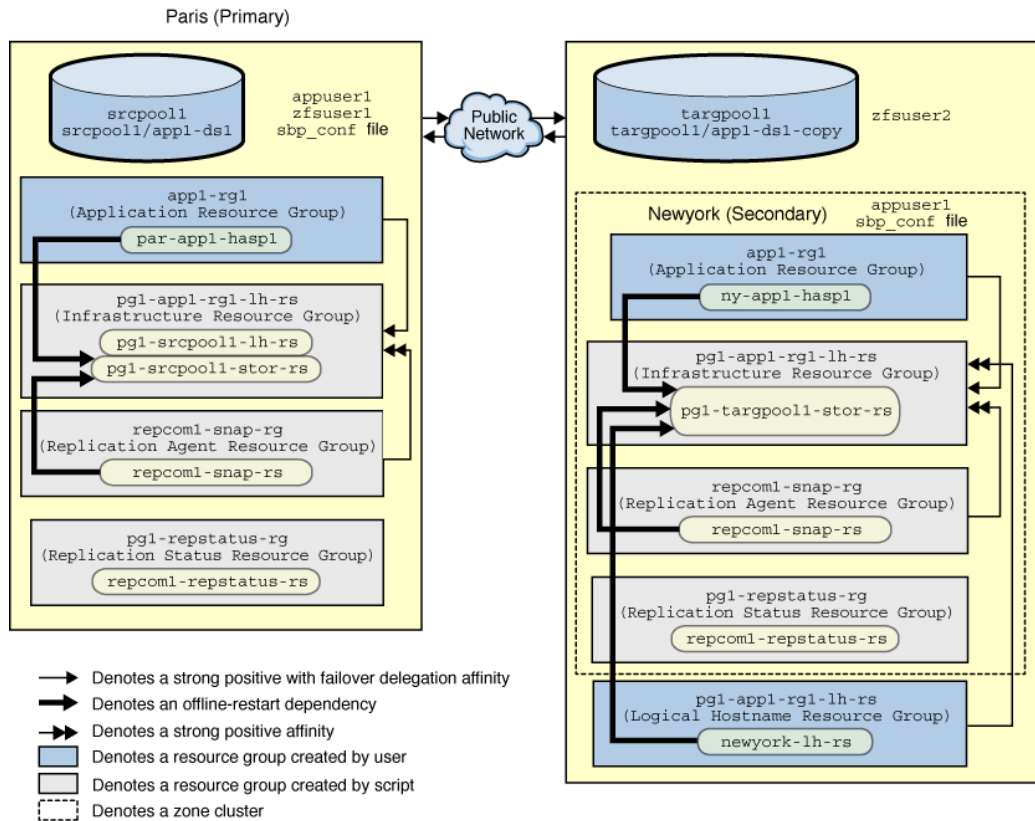
11. Type the following command from one node of either partner cluster to confirm whether the protection group started on both clusters.

```
# geopg status pg1
```

Use Case: Configuring Oracle Solaris ZFS Snapshot Replication With One Partner Cluster as Zone Cluster

This example shows how to set up the protection group with Oracle Solaris ZFS snapshot replication to protect and manage the application and its ZFS datasets with one partner being a zone cluster. Assume that the application resource group and application user is already set up. If a partner cluster is a zone cluster, execute the commands for setting up logical hostname, private string, and ZFS permissions in the global zone of the nodes of the zone cluster. Any disaster recovery framework commands must be executed within the zone cluster. If either partner cluster is a zone cluster, you must create the private string to store the SSH passphrase on both the partners. You must create logical hostname resource and resource group in the global zone of the zone cluster partner. Assume that the primary cluster `paris` is a global cluster and secondary cluster `newyork` is a zone cluster. The following figure shows the setup of resource groups and resources for Oracle Solaris ZFS Snapshot Replication With One Partner Cluster as Zone Cluster.

FIGURE 3 Example Resource Group and Resource Setup With One Partner Cluster as Global Zone and Other Partner Cluster as Zone Cluster



Perform the following actions to configure the Oracle Solaris ZFS snapshot replication:

1. Decide the hostname to use on each cluster as replication-related logical hostname. For this example, assume that the logical hostname on the paris cluster is paris-lh. The logical hostname on the newyork cluster is newyork-lh.

Add the logical hostname entries for paris-lh in the /etc/hosts file on both the nodes of the paris cluster. Similarly, add the logical hostname entries for newyork-lh in the /etc/hosts file in the global zone on both nodes of the newyork cluster.

2. Create the replication user and SSH setup. For information about the step for creating the replication user and SSH setup, see [“Use Case: Setting Up Replication User and SSH” on page 100](#).
3. Provide ZFS permissions to replication user on both clusters. You must import the ZFS storage pool to provide permissions. If application's HAStoragePlus is online, log in as the root user on the node where it is online. Else if the ZFS storage pool is not online on any cluster node, import the ZFS storage pool on any node and perform ZFS commands on that node. On each cluster, log in as the root user to the cluster node where the zpool is imported.

Type the following commands in the global zone of the paris cluster node where srcpool1 is imported:

```
# /sbin/zfs allow zfsuser1
  create,destroy,hold,mount,receive,release,send,rollback,snapshot \
srcpool1/app1-ds1
# /sbin/zfs allow srcpool1/app1-ds1
---- Permissions on srcpool1/app1-ds1 -----
Local+Descendent permissions:
user zfsuser1 create,destroy,hold,mount,receive,release,rollback,send,snapshot
#
```

Type the following commands in the global zone of the newyork cluster node where the targpool1 zpool is imported.

```
# /sbin/zfs allow zfsuser2
  create,destroy,hold,mount,receive,release,send,rollback,snapshot \
targpool1/app1-ds1-copy
# /sbin/zfs allow targpool1/app1-ds1-copy
---- Permissions on targpool1/app1-ds1-copy -----
Local+Descendent permissions:
user zfsuser2 create,destroy,hold,mount,receive,release,rollback,send,snapshot
#
```

Note that the replication user on each cluster must have the above Local+Descendent ZFS permissions on the ZFS dataset used on that cluster.

4. Since newyork is a zone cluster, create Oracle Solaris Cluster private strings in the global zone of each partner to store the SSH passphrase for the replication user of that partner. The private string object name must be in the format *zonename:replication-component:local_service_passphrase*.

Since the paris cluster is not a zone cluster, therefore you must use global as the zonename. Type the following command in any one node of the paris cluster to create the private string for zfsuser1, and specify the SSH passphrase for zfsuser1 on the prompt:

```
# /usr/cluster/bin/clpstring create -b global:repcom1:local_service_passphrase \
global:repcom1:local_service_passphrase
Enter string value:
Enter string value again:
```

Suppose newyork is the zone cluster name. The zone cluster name is always the same as the zone name by restriction, so you must use newyork as the zonename. Type the following command in the global zone of any one zone of the newyork cluster to create the private string for zfsuser2, and specify the SSH passphrase for zfsuser2 on the prompt :

```
# /usr/cluster/bin/clpstring create -b newyork:repcom1:local_service_passphrase \
newyork:repcom1:local_service_passphrase
Enter string value:
Enter string value again:
```

- Suppose you create a file `/var/tmp/geo/zfs_snapshot/sbp_conf` file to use as the script-based plugin configuration file on both clusters to specify replication component node lists and evaluation rules. Add the following entry in the `/var/tmp/geo/zfs_snapshot/sbp_conf` file on each node of the paris cluster :

```
repcom1|any|paris-node-1,paris-node-2
```

Add the following entry in the `/var/tmp/geo/zfs_snapshot/sbp_conf` file in each zone of the newyork zone cluster :

```
repcom1|any|newyork-node-1,newyork-node-2
```

- Copy `/opt/ORCLscgprepzfssnap/etc/zfs_snap_geo_config` to create a parameters file `/var/tmp/geo/zfs_snapshot/repcom1_conf` for the replication component from one node of primary cluster paris. Type the configuration parameters in the configuration file.

```
PS=paris-newyork
PG=pg1
REPCOMP=repcom1
REPRS=repcom1-repstatus-rs
REPRG=pg1-repstatus-rg
DESC="Protect app1-rg1 using ZFS snapshot replication"
APPRG=app1-rg1
CONFIGFILE=/var/tmp/geo/zfs_snapshot/sbp_conf
LOCAL_REP_USER=zfsuser1
REMOTE_REP_USER=zfsuser2
LOCAL_PRIV_KEY_FILE=
REMOTE_PRIV_KEY_FILE=/export/home/zfsuser2/.ssh/zfsrep1
LOCAL_ZPOOL_RS=par-app1-hasp1
REMOTE_ZPOOL_RS=ny-app1-hasp1
```

```

LOCAL_LH=paris-lh
REMOTE_LH=newyork-lh
LOCAL_DATASET=srcpool1/app1-ds1
REMOTE_DATASET=targpool1/app1-ds1-copy
REPLICATION_INTERVAL=120
NUM_OF_SNAPSHOTS_TO_STORE=2
REPLICATION_STREAM_PACKAGE=false
SEND_PROPERTIES=true
INTERMEDIARY_SNAPSHOTS=false
RECURSIVE=true
MODIFY_PASSPHRASE=false

```

7. Execute the `zfs_snap_geo_register` script using the replication configuration file as parameter on any one node of the primary cluster `paris`. You must perform this step from the same node as in Step 6.

```

$ /opt/ORCLscgprepzfssnap/util/zfs_snap_geo_register -f /var/tmp/geo/zfs_snapshot/
repcom1_conf

```

The setup actions by `zfs_snap_geo_register` script creates the following components in the primary cluster, as shown in [Figure 3, “Example Resource Group and Resource Setup With One Partner Cluster as Global Zone and Other Partner Cluster as Zone Cluster,”](#) on [page 114](#):

- Protection group `pg1`
 - Replication component `repcom1`
 - Infrastructure resource group `pg1-srcpool1-infr-rg` and its resources
 - Replication resource group `repcom1-snap-rg` which contains the resource `repcom1-snap-rs`
 - Replication status resource group `pg1-repstatus-rg` and replication status resource `repcom1-repstatus-rs`
8. On any node of the `paris` cluster, check that the protection group and replication component is created successfully.

```

# /usr/cluster/bin/geopg show pg1
# /usr/cluster/bin/geopg status pg1

```

9. On any one zone of the `newyork` cluster, get the protection group that is created on the primary cluster. You must type the following command in the zone cluster.

```

# /usr/cluster/bin/geopg get -s paris-newyork pg1

```

This command creates the resource group and resource setup in the zone cluster which is the secondary cluster, as shown in [Figure 3, “Example Resource Group and Resource Setup](#)

With One Partner Cluster as Global Zone and Other Partner Cluster as Zone Cluster,” on page 114.

10. Configure a logical hostname resource and resource group in the global zone of the zone cluster partner newyork to host the replication hostname. Type the following commands on one node of the global zone.

The zone cluster partner is newyork, and replication hostname to use for newyork is newyork-lh. The infrastructure resource group created automatically in the newyork zone cluster is pg1-targpool1-infr-rg. The storage resource in that infrastructure resource group is pg1-targpool1-stor-rs. Type the following commands in the global zone of any one zone of newyork cluster:

```
# clrg create newyork-lh-rg
# clrslh create -g newyork-lh-rg -h newyork-lh newyork-lh-rs
# clrg manage newyork-lh-rg
# clrg set -p RG_affinities===newyork:pg1-targpool1-infr-rg newyork-lh-rg
(C538594) WARNING: resource group global:newyork-lh-rg has a strong positive affinity
on resource group
newyork:pg1-targpool1-infr-rg with Auto_start_on_new_cluster=FALSE; global:newyork-
lh-rg will
be forced to remain offline until its strong affinities are satisfied.
# clrs set -p Resource_dependencies_offline_restart=newyork:pg1-targpool1-stor-rs
newyork-lh-rs
# clrg show -p Auto_start_on_new_cluster newyork-lh-rg
=== Resource Groups and Resources ===
Resource Group:          newyork-lh-rg
Auto_start_on_new_cluster: True
```

If the Auto_start_on_new_cluster property is not set to True, type the following command:

```
# clrg set -p Auto_start_on_new_cluster=True newyork-lh-rg
```

11. From any zone of the newyork cluster, check that the protection group and replication components are available. Ensure that the protection group synchronization status between paris and newyork cluster shows OK. You must type the following command in the zone cluster:

```
$ /usr/cluster/bin/geopg show pg1
$ /usr/cluster/bin/geopg status pg1
```

Similarly, check status from any one node of paris cluster:

```
$ /usr/cluster/bin/geopg show pg1
$ /usr/cluster/bin/geopg status pg1
```

12. Activate the protection group to start the Oracle Solaris ZFS snapshot replication from any node of the `paris` or `newyork` cluster.

```
$ /usr/cluster/bin/geopg start -e global pg1
```

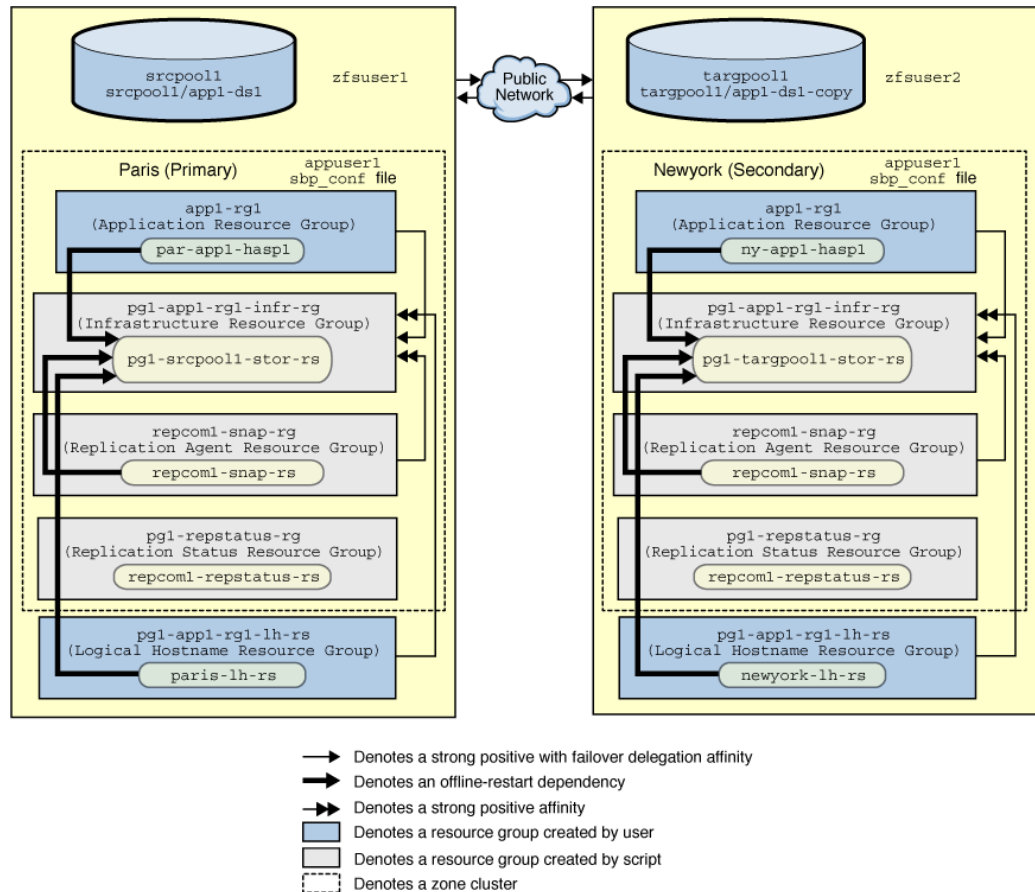
13. Type the following command from one node of either partner cluster to confirm whether the protection group started on both clusters.

```
# geopg status pg1
```

Use Case: Configuring Oracle Solaris ZFS Snapshot Replication With Both Partner Clusters as Zone Cluster

This example shows how to set up the protection group with Oracle Solaris ZFS snapshot replication to protect and manage the application and its ZFS datasets with both the partner clusters as zone cluster. Assume that the application resource group and application user is already set up. Execute the commands for setting up logical hostname, private string and ZFS permissions in the global zone of the nodes of the zone cluster. Any disaster recovery framework commands must be executed within the zone cluster. You must create the private string to store the SSH passphrase on both the partners. You must create logical hostname resource and resource group in the global zone of the zone cluster partner. Suppose that both the primary cluster `paris` and the secondary cluster `newyork` are zone clusters.

FIGURE 4 Example Resource Group and Resource Setup for Oracle Solaris ZFS Snapshot Replication With Both Partner Cluster as Zone Cluster



Perform the following actions to configure the Oracle Solaris ZFS snapshot replication:

1. Decide the hostname to use on each cluster as replication-related logical hostname. For this example, assume that the logical hostname on the paris cluster is paris-lh. The logical hostname on the newyork cluster is newyork-lh.

Add the logical hostname entries for paris-lh in the /etc/hosts file in the global zone on both nodes of the paris cluster. Similarly, add the logical hostname entries for newyork-lh in the /etc/hosts file in the global zone on both nodes of the newyork cluster.

2. Create the replication user and SSH setup. For information about the step for creating the replication user and SSH setup, see [“Use Case: Setting Up Replication User and SSH” on page 100](#).
3. Provide ZFS permissions to replication user on both clusters. You must import the ZFS storage pool to provide permissions. If application's HAStoragePlus is online, log in as root user on the node where it is online. Else if the ZFS storage pool is not online on any cluster node, import the ZFS storage pool on any node and perform ZFS commands on that node. On each cluster, log in as the root user to the cluster node where the zpool is imported.

Type the following commands in the global zone of the node of the paris cluster where the srcpool1 zpool is imported.

```
# /sbin/zfs allow zfsuser1
  create,destroy,hold,mount,receive,release,send,rollback,snapshot \
srcpool1/app1-ds1
# /sbin/zfs allow srcpool1/app1-ds1
---- Permissions on srcpool1/app1-ds1 -----
Local+Descendent permissions:
user zfsuser1 create,destroy,hold,mount,receive,release,rollback,send,snapshot
#
```

Type the following commands in the global zone of the node of the newyork cluster where the targpool1 zpool is imported.

```
# /sbin/zfs allow zfsuser2
  create,destroy,hold,mount,receive,release,send,rollback,snapshot \
targpool1/app1-ds1-copy
# /sbin/zfs allow targpool1/app1-ds1-copy
---- Permissions on targpool1/app1-ds1-copy -----
Local+Descendent permissions:
user zfsuser2 create,destroy,hold,mount,receive,release,rollback,send,snapshot
#
```

Note that the replication user on each cluster must have the above Local+Descendent ZFS permissions on the ZFS dataset used on that cluster.

4. Since newyork is a zone cluster, create Oracle Solaris Cluster private strings in the global zone of each partner to store the SSH passphrase for the replication user of that partner. The private string object name must be in the format *zonename:replication-component:local_service_passphrase*. The zone cluster name is always the same as the zone name by restriction.

Since paris cluster is also a zone cluster, therefore you must use paris as the zonename. Type the following command in the global zone of any one zone of the paris cluster to create the private string for zfsuser1, and specify the SSH passphrase for zfsuser1 on the prompt:

```
# /usr/cluster/bin/clpstring create -b paris:repcom1:local_service_passphrase \
paris:repcom1:local_service_passphrase
Enter string value:
Enter string value again:
```

Since newyork is the zone cluster name, you must use newyork as the zonename. Type the following command in the global zone of any one zone of the newyork cluster to create the private string for zfsuser2, and specify the SSH passphrase for zfsuser2 on the prompt :

```
# /usr/cluster/bin/clpstring create -b newyork:repcom1:local_service_passphrase \
newyork:repcom1:local_service_passphrase
Enter string value:
Enter string value again:
```

- Suppose you create a file `/var/tmp/geo/zfs_snapshot/sbp_conf` to use as the script-based plugin configuration file on both clusters to specify replication component node lists and evaluation rules. Add the following entry in the `/var/tmp/geo/zfs_snapshot/sbp_conf` file in each zone of the paris cluster :

```
repcom1|any|paris-node-1,paris-node-2
```

Add the following entry in the `/var/tmp/geo/zfs_snapshot/sbp_conf` file in each zone of the newyork zone cluster :

```
repcom1|any|newyork-node-1,newyork-node-2
```

- Copy `/opt/ORCLscgprepzfsnap/etc/zfs_snap_geo_config` to create a parameters file `/var/tmp/geo/zfs_snapshot/repcom1_conf` for the replication component from one node of the primary cluster paris. Type the configuration parameters in the configuration file.

```
PS=paris-newyork
PG=pg1
REPCOMP=repcom1
REPRS=repcom1-repstatus-rs
REPRG=pg1-repstatus-rg
DESC="Protect appl-rg1 using ZFS snapshot replication"
APPRG=appl-rg1
CONFIGFILE=/var/tmp/geo/zfs_snapshot/sbp_conf
LOCAL_REP_USER=zfsuser1
REMOTE_REP_USER=zfsuser2
LOCAL_PRIV_KEY_FILE=
REMOTE_PRIV_KEY_FILE=/export/home/zfsuser2/.ssh/zfsrep1
LOCAL_ZPOOL_RS=par-app1-haspl
REMOTE_ZPOOL_RS=ny-app1-haspl
LOCAL_LH=paris-lh
```

```

REMOTE_LH=newyork-lh
LOCAL_DATASET=srcpool1/app1-ds1
REMOTE_DATASET=targpool1/app1-ds1-copy
REPLICATION_INTERVAL=120
NUM_OF_SNAPSHOTS_TO_STORE=2
REPLICATION_STREAM_PACKAGE=false
SEND_PROPERTIES=true
INTERMEDIARY_SNAPSHOTS=false
RECURSIVE=true
MODIFY_PASSPHRASE=false

```

7. Execute the `zfs_snap_geo_register` script using the replication configuration file as parameter on any one zone of the primary cluster `paris`. You must perform the step from the same node as in Step 6.

```

# /opt/ORCLscgprepzfssnap/util/zfs_snap_geo_register -f /var/tmp/geo/zfs_snapshot/
repcom1_conf

```

The set up actions performed by the `zfs_snap_geo_register` script creates the following components in the primary cluster, as shown in [Figure 4, “Example Resource Group and Resource Setup for Oracle Solaris ZFS Snapshot Replication With Both Partner Cluster as Zone Cluster,”](#) on page 120:

- Protection group `pg1`
 - Replication component `repcom1`
 - Infrastructure resource group `pg1-srcpool1-infr-rg` and the storage resource `pg1-srcpool1-stor-rs`
 - Replication resource group `repcom1-snap-rg` which contains the resource `repcom1-snap-rs`
 - Replication status resource group `pg1-repstatus-rg` and replication status resource `repcom1-repstatus-rs`
8. From any zone of the `paris` cluster, check that the protection group and replication component is created successfully.

```

# /usr/cluster/bin/geopg show pg1
# /usr/cluster/bin/geopg status pg1

```

9. Configure a logical hostname resource and resource group in the global zone of the zone cluster partner `paris` to host the replication hostname. Type the following commands on one node of the global zone.

The zone cluster partner is `paris`, and replication hostname to use for `paris` is `paris-lh`. The infrastructure resource group created automatically in the `paris` zone cluster is `pg1-srcpool1-infr-rg`. The storage resource in that infrastructure resource group is `pg1-`

srcpool1-stor-rs. Type the following commands in the global zone of any one zone of paris cluster:

```
# clrg create paris-lh-rg
# clrslh create -g paris-lh-rg -h paris-lh paris-lh-rs
# clrg manage paris-lh-rg
# clrg set -p RG_affinities=++paris:pg1-srcpool1-infr-rg paris-lh-rg
(C538594) WARNING: resource group global:paris-lh-rg has a strong positive affinity
on resource group
paris:pg1-srcpool1-infr-rg with Auto_start_on_new_cluster=FALSE; global:paris-lh-rg
will
be forced to remain offline until its strong affinities are satisfied.
# clrs set -p Resource_dependencies_offline_restart=paris:pg1-srcpool1-stor-rs paris-
lh-rs
# clrg show -p Auto_start_on_new_cluster paris-lh-rg
=== Resource Groups and Resources ===
Resource Group:          paris-lh-rg
Auto_start_on_new_cluster: True
```

If the `Auto_start_on_new_cluster` property is not set to `True`, type the following command:

```
# clrg set -p Auto_start_on_new_cluster=True paris-lh-rg
```

10. On any one zone of the newyork zone cluster, get the protection group that is created on the primary cluster.

```
# /usr/cluster/bin/geopg get -s paris-newyork pg1
```

This command creates the resource setup in the secondary cluster.

11. Configure a logical hostname resource and resource group in the global zone of the zone cluster partner newyork to host the replication hostname. Type the following commands on one node of the global zone.

The zone cluster partner is newyork, and replication hostname to use for newyork is newyork-lh. The infrastructure resource group created automatically in the newyork zone cluster is `pg1-targpool1-infr-rg`. The storage resource in that infrastructure resource group is `pg1-targpool1-stor-rs`. Type the following commands in the global zone of any one zone of newyork cluster:

```
# clrg create newyork-lh-rg
# clrslh create -g newyork-lh-rg -h newyork-lh newyork-lh-rs
# clrg manage newyork-lh-rg
# clrg set -p RG_affinities=++newyork:pg1-targpool1-infr-rg newyork-lh-rg
(C538594) WARNING: resource group global:newyork-lh-rg has a strong positive affinity
on resource group
```

```
newyork:pg1-targpool1-infr-rg with Auto_start_on_new_cluster=FALSE; global:newyork-
lh-rg will
be forced to remain offline until its strong affinities are satisfied.
# clrs set -p Resource_dependencies_offline_restart=newyork:pg1-targpool1-stor-rs
newyork-lh-rs
# clrg show -p Auto_start_on_new_cluster newyork-lh-rg
=== Resource Groups and Resources ===
Resource Group:          newyork-lh-rg
Auto_start_on_new_cluster: True
```

If the `Auto_start_on_new_cluster` property is not set to `True`, type the following command:

```
# clrg set -p Auto_start_on_new_cluster=True newyork-lh-rg
```

12. From any one zone of the `newyork` zone cluster, check that the protection group and replication components are available. Ensure that the protection group synchronization status between `paris` and `newyork` cluster shows OK.

```
# /usr/cluster/bin/geopg show pg1
# /usr/cluster/bin/geopg status pg1
```

Similarly, execute such commands in any one zone of `paris` zone cluster to check status:

```
# /usr/cluster/bin/geopg show pg1
# /usr/cluster/bin/geopg status pg1
```

13. Activate the protection group from any one zone of the `paris` zone cluster to start the Oracle Solaris ZFS snapshot replication.

```
# /usr/cluster/bin/geopg start -e global pg1
```

14. Type the following command from one node of either partner cluster to confirm whether the protection group started on both clusters.

```
# geopg status pg1
```

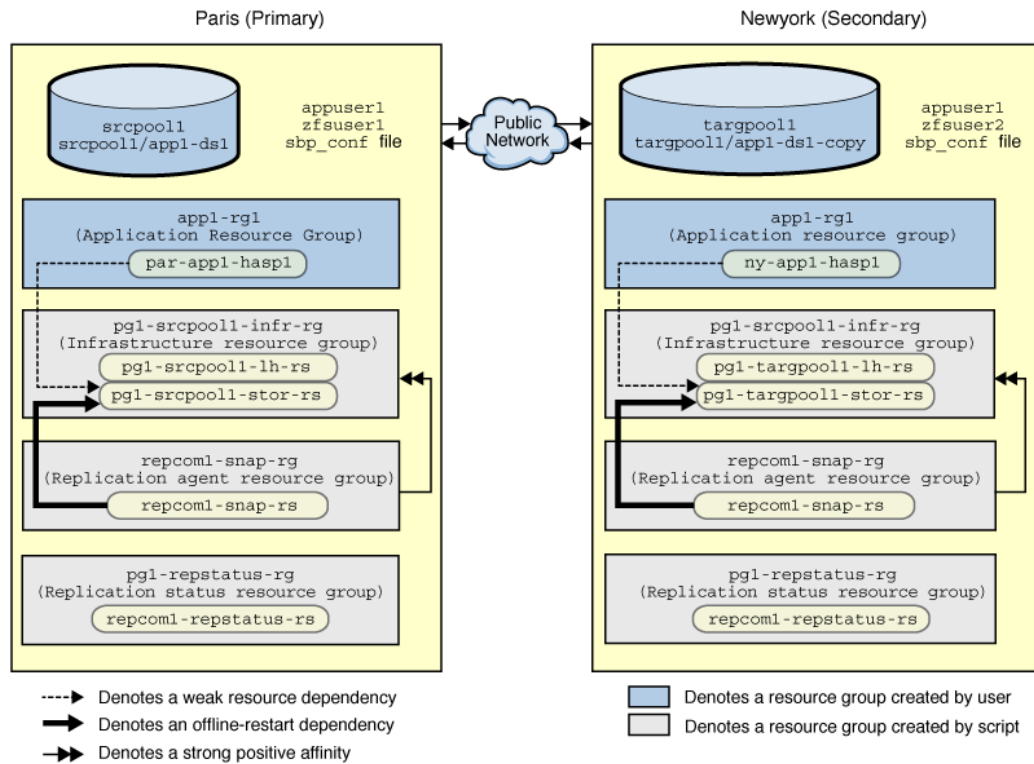
Use Case: Configuring Oracle Solaris ZFS Snapshot Replication With Zpools for Globally Mounted ZFS Filesystems

If you are using zpools for globally mounted ZFS filesystems to configure Oracle Solaris ZFS snapshot replication, the configuration is similar to the use [“Use Case: Configuring Oracle Solaris ZFS Snapshot Replication When Both Partner Clusters Are Global”](#)

Zones” on page 108. As the zpool is global, you would have already configured the device group and is visible with the `cldg list` command. The application resource group would already have an `HASStoragePlus` resource to manage the zpool used by the application. The replication module does not use the `ZpoolsExportOnStop` property on the resource.

An application can be configured as failover or scalable with the zpool for globally mounted ZFS filesystems.

FIGURE 5 Scalable Application Configured with Zpool for Globally Mounted ZFS Filesystems



You can perform the following steps to configure the Oracle Solaris ZFS snapshot replication by using the `zfs_snap_geo_register` script:

1. Decide the hostname to use on each cluster as replication-related logical hostname. For this example, assume that the logical hostname on the paris cluster is `paris-lh`. The logical hostname on the newyork cluster is `newyork-lh`.

Add the logical hostname entries for `paris-lh` in the `/etc/hosts` file on both the nodes of the `paris` cluster. Similarly, add the logical hostname entries for `newyork-lh` in the `/etc/hosts` file on both the nodes of the `newyork` cluster.

2. Create the replication user and SSH setup. For information about the step for creating the replication user and SSH setup, see [“Use Case: Setting Up Replication User and SSH” on page 100](#).
3. Provide ZFS permissions to replication user on both clusters. You must import the ZFS storage pool to provide permissions. If the zpool is imported, log in as the root user on the node where the zpool is imported. Else if the zpool is not imported on any cluster node, import the zpool using the `cldg online zpoolname` command and type the ZFS commands on that node.

Type the following commands on the node of the `paris` cluster where `srcpool1` is imported:

```
# /sbin/zfs allow zfsuser1
  create,destroy,hold,mount,receive,release,send,rollback,snapshot \
srcpool1/app1-ds1
# /sbin/zfs allow srcpool1/app1-ds1
---- Permissions on srcpool1/app1-ds1 -----
Local+Descendent permissions:
user zfsuser1 create,destroy,hold,mount,receive,release,rollback,send,snapshot
#
```

Type the following commands on the node of the `newyork` cluster where `targpool1` is imported:

```
# /sbin/zfs allow zfsuser2
  create,destroy,hold,mount,receive,release,send,rollback,snapshot \
targpool1/app1-ds1-copy
# /sbin/zfs allow targpool1/app1-ds1-copy
---- Permissions on targpool1/app1-ds1-copy -----
Local+Descendent permissions:
user zfsuser2 create,destroy,hold,mount,receive,release,rollback,send,snapshot
#
```

Note that the replication user on each cluster must have the above `Local+Descendent` ZFS permissions on the ZFS dataset used on that cluster. If you brought the device group online, take it offline using the `cldg offline zpoolname` command.

4. Suppose you create a file `/var/tmp/geo/zfs_snapshot/sbp_conf` file to use as the script-based plugin configuration file on both clusters to specify replication component node lists and evaluation rules. Add the following entry in the `/var/tmp/geo/zfs_snapshot/sbp_conf` file on each node of the `paris` cluster :

```
repcom1|any|paris-node-1,paris-node-2
```

Add the following entry in the `/var/tmp/geo/zfs_snapshot/sbp_conf` file on each node of the newyork cluster :

```
repcom1|any|newyork-node-1,newyork-node-2
```

5. Copy `/opt/ORCLscgrepzfssnap/etc/zfs_snap_geo_config` to create a parameters file `/var/tmp/geo/zfs_snapshot/repcom1_conf` for the replication component. Type the configuration parameters in the configuration file.

```
PS=paris-newyork
PG=pg1
REPCOMP=repcom1
REPRS=repcom1-repstatus-rs
REPRG=pg1-repstatus-rg
DESC="Protect app1-rg1 using ZFS snapshot replication"
APPRG=app1-rg1
CONFIGFILE=/var/tmp/geo/zfs_snapshot/sbp_conf
LOCAL_REP_USER=zfsuser1
REMOTE_REP_USER=zfsuser2
LOCAL_PRIV_KEY_FILE=
REMOTE_PRIV_KEY_FILE=/export/home/zfsuser2/.ssh/zfsrep1
LOCAL_ZPOOL_RS=par-app1-hasp1
REMOTE_ZPOOL_RS=ny-app1-hasp1
LOCAL_LH=paris-lh
REMOTE_LH=newyork-lh
LOCAL_DATASET=srcpool1/app1-ds1
REMOTE_DATASET=targpool1/app1-ds1-copy
REPLICATION_INTERVAL=120
NUM_OF_SNAPSHOTS_TO_STORE=2
REPLICATION_STREAM_PACKAGE=false
SEND_PROPERTIES=true
INTERMEDIARY_SNAPSHOTS=false
RECURSIVE=true
MODIFY_PASSPHRASE=false
```

6. Execute the `zfs_snap_geo_register` script using the replication configuration file as parameter from any one node of paris cluster. specify the SSH passphrase for `zfsuser1` of paris cluster at the Password for property `local_service_password` prompt and the SSH passphrase for `zfsuser2` of newyork cluster at the Password for property `remote_service_password` prompt

```
# /opt/ORCLscgrepzfssnap/util/zfs_snap_geo_register -f /var/tmp/geo/zfs_snapshot/repcom1_conf
```


Password for property local_service_password :
Password for property remote_service_password :

This setup action performed by the `zfs_snap_geo_register` script creates the following components in the primary cluster, as shown in [Figure 2, “Example Setup of Oracle Solaris ZFS Snapshot Replication in the Global Zone on Both the Partner Clusters,”](#) on page 109:

- Protection group `pg1`
- Replication component `repcom1`
- Infrastructure resource group `pg1-srcpool1-infr-rg` and its resources as shown in the figure
- Replication resource group `repcom1-snap-rg` which contains the resource `repcom1-snap-rs`
- Replication status resource group `pg1-repstatus-rg` and replication status resource `repcom1-repstatus-rs`

7. On any node of the `paris` cluster, check that the protection group and replication component is created successfully.

```
# /usr/cluster/bin/geopg show pg1  
# /usr/cluster/bin/geopg status pg1
```

8. On any node of the `newyork` cluster, get the protection group that is created on the primary cluster.

```
# /usr/cluster/bin/geopg get -s paris-newyork pg1
```

This command creates the resource groups and resources in the secondary cluster, as shown in [Figure 2, “Example Setup of Oracle Solaris ZFS Snapshot Replication in the Global Zone on Both the Partner Clusters,”](#) on page 109.

9. From any node of the `newyork` cluster, check that the protection group and replication components are available. Ensure that the protection group synchronization status between `paris` and `newyork` cluster shows OK.

```
# /usr/cluster/bin/geopg show pg1  
# /usr/cluster/bin/geopg status pg1
```

Similarly, check status from any node of `paris` cluster:

```
# /usr/cluster/bin/geopg show pg1  
# /usr/cluster/bin/geopg status pg1
```

10. Activate the protection group to start the Oracle Solaris ZFS snapshot replication.

```
$ /usr/cluster/bin/geopg start -e global pg1
```

11. Type the following command from one node of either partner cluster to confirm whether the protection group started on both clusters.

```
# geopg status pg1
```

Note - With zpools for globally mounted ZFS filesystems, an application can be configured as failover or scalable. For failover applications, administration steps remain the same. Only resource group and resource structure changes. The Oracle Solaris Cluster Geographic Edition software sets a strong positive affinity with failover delegation from the application resource group to the infrastructure resource group and an offline-restart resource dependency from the application's HASStoragePlus resource managing the zpool to the replication infrastructure HASStoragePlus resource for the same zpool.

Configuring Oracle Solaris ZFS Snapshot Replication for ZFS-based Cluster Filesystems in Zone Clusters

To configure ZFS snapshot replication for ZFS-based cluster filesystems in zone clusters, the ZFS snapshot protection groups must run on the global clusters. This is because the zpool for globally mounted filesystems is only available in the global cluster. The ZFS-based cluster filesystem is loopback mounted to the zone cluster nodes and these zone clusters will be in a partnership with protection group of type NONE.

This appendix contains the following sections:

- [“Configuration Requirements” on page 131](#)
- [“Use Case: Configuring Oracle Solaris ZFS Snapshot Replication for ZFS-based Cluster Filesystems in Zone Clusters” on page 132](#)

Configuration Requirements

1. Partnership between global clusters containing ZFS snapshot replication protection group to manage replication of the filesystem(s) in zpool for globally mounted filesystems.
2. On each global cluster, the ZFS-based cluster filesystem is loopback mounted to zone cluster. A partnership between these zone clusters with protection group of type NONE for managing the application using the ZFS-based cluster filesystem in the zone clusters.
3. Application resource group on the global clusters with HAStoragePlus resource that contains zpool for globally mounted filesystems in its `globalzpools` property. This resource group is added to ZFS snapshot replication protection group in the global clusters.
4. Application resource group in the zone clusters with HAStoragePlus resource containing loopback mounted filesystem(s) in its `filesystemmountpoints` property. This resource group has `++ rg_affinity` with the resource group (from item 3 above) on its global cluster. The HAStoragePlus resource in this

resource contains global cluster HAStoragePlus resource (from item 3 above) in its `resource_dependencies_offline_restart` property.

5. On each site a disaster recovery framework site is created that contains the global cluster and the zone cluster in the site.
6. A multigroup is added to each site that contains the global cluster protection group and the zone cluster protection group. The zone cluster protection group has a dependency on the global cluster protection group.
7. To manage start, stop, switchover, and takeover of the configuration, use the `geomg` command on the multigroup.

The following use case illustrates the procedure for bringing up this configuration.

Use Case: Configuring Oracle Solaris ZFS Snapshot Replication for ZFS-based Cluster Filesystems in Zone Clusters

The focus of this procedure is the illustration of steps required for ZFS-based cluster filesystems in zone clusters replicated with ZFS snapshot replication. As such, it does not show setting up replication users, nor SSH and other steps, such as preparing a zone cluster for the disaster recovery framework. Such procedures are found elsewhere in this guide and other disaster recovery framework guides. For information about the step for creating the replication user and SSH setup, see [“Use Case: Setting Up Replication User and SSH” on page 100](#)

The example uses the following names:

- Global cluster `paris` with nodes `phys-paris-1` and `phys-paris-2`
- Zone cluster `paris-zc` on `paris` with nodes `paris-zc-1` and `paris-zc-2`
- Global cluster `newyork` with nodes `phys-newyork-1` and `phys-newyork-2`
- Zone cluster `newyork-zc` on `newyork` with nodes `newyork-zc-1` and `newyork-zc-2`
- On `paris` cluster the zpool is named `dataz4` with `dataz4/fs1` being replicated
- On `newyork` cluster the zpool is named `dataz5` with `dataz5/fs1` being replicated
- `paris-rep-lh` is replication logical host on `paris` cluster
- `newyork-rep-lh` is replication logical host on `newyork` cluster

This example contains the procedure required for this configuration. It shows scalable Apache for illustration purposes only.

1. From one node of `paris` global cluster create zpool, filesystem and configure it in HAStoragePlus resource setting `globalzpools` property.

```

root@phys-paris-1:~# zpool create dataz4 c0t600144F08010407600005BAEA4BC0004d0
root@phys-paris-1:~# zfs create dataz4/fs1
root@phys-paris-1:~# clrg create -S -p auto_start_on_new_cluster=false gzfs-gc-rg
root@phys-paris-1:~# cldg list dataz4
cldg: (C531091) Device group "dataz4" does not exist.
root@phys-paris-1:~# zpool export dataz4
root@phys-paris-1:~# clrs create -g gzfs-gc-rg -t HASStoragePlus -p \
globalzpools=dataz4 hasp4dataz4
root@phys-paris-1:~# cldg status dataz4

```

=== Cluster Device Groups ===

--- Device Group Status ---

Device Group Name	Primary	Secondary	Status
dataz4	-	-	Offline

```

root@phys-paris-1:~# cldg show dataz4

```

=== Device Groups ===

```

Device Group Name:          dataz4
Type:                       ZPOOL
failback:                   false
Node List:                   phys-paris-2, phys-paris-1
preferenced:                 false
autogen:                     true
numsecondaries:              1
ZFS pool name:               dataz4
poolaccess:                  global
readonly:                    false
import-at-boot:              false
searchpaths:                 /dev/dsk

```

```

root@phys-paris-1:~# clrg online -M gzfs-gc-rg

```

```

root@phys-paris-1:~# cldg status dataz4

```

=== Cluster Device Groups ===

--- Device Group Status ---

Device Group Name	Primary	Secondary	Status
dataz4	-	-	Offline

```

dataz4                phys-paris-2  phys-paris-1  Online

root@phys-paris-1:~# df -h | grep dataz4
dataz4                14G   46K    14G   1%   /dataz4
dataz4/fs1            14G   40K    14G   1%   /dataz4/fs1
root@phys-paris-1:~# zpool list dataz4
cannot open 'dataz4': no such pool
root@phys-paris-1:~#

```

The zpool is imported on one node of the global cluster but the filesystem is mounted on both global cluster nodes.

```

root@phys-paris-2:~# df -h | grep dataz4
dataz4                20G   46K    14G   1%   /dataz4
dataz4/fs1            20G   40K    14G   1%   /dataz4/fs1
root@phys-paris-2:~# zpool list dataz4
NAME  SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
dataz4 19.9G  158K  19.9G  0%  1.00x  ONLINE  /
root@phys-paris-2:~#

```

2. Configure the ZFS cluster file system in the zone cluster and create resource group and HAStoragePlus resource for it setting `rg_affinities` and `resource_dependencies_offline_restart` properties.

```

root@phys-paris-1:~# clzc configure paris-zc
clzc:paris-zc> add fs
clzc:paris-zc:fs> set dir=/global/apache
clzc:paris-zc:fs> set special=/dataz4/fs1
clzc:paris-zc:fs> set type=lofs
clzc:paris-zc:fs> end
clzc:paris-zc> exit
root@phys-paris-1:~# clrg create -Z paris-zc -S -p rg_affinities=++global:gzfsrg \
-p auto_start_on_new_cluster=false apache-rg
root@phys-paris-1:~# clrs create -Z paris-zc -t HAStoragePlus -p \
filesystemmountpoints=/global/apache -p \
resource_dependencies_offline_restart=global:hasp4dataz4 -g apache-rg hasp4apache
root@phys-paris-1:~# clrg online -M -Z paris-zc apache-rg
root@phys-paris-1:~# zlogin paris-zc
[Connected to zone 'paris-zc' pts/3]
Oracle Corporation      SunOS 5.11      11.4   September 2018
You have new mail.
root@paris-zc-1:~# df -h | grep apache
/global/apache        20G   40K    20G   1%   /global/apache
root@paris-zc-1:~#

```

The filesystem is mounted on both zone cluster nodes.

```

root@phys-paris-2:~# zlogin paris-zc
[Connected to zone 'paris-zc' pts/4]
Oracle Corporation      SunOS 5.11      11.4    September 2018
You have new mail.
root@paris-zc-2:~# df -h | grep apache
/global/apache          20G   40K      20G    1%   /global/apache
root@paris-zc-2:~#

```

3. From one node of newyork global cluster create zpool, filesystem and configure it in HAStoragePlus resource setting globalzpools property.

```

root@phys-newyork-1:~# zpool create dataz5 c0t600144F0CD152C9E00005BAE9BB90009d0
root@phys-newyork-1:~# zfs create dataz5/fs1
root@phys-newyork-1:~# clrg create -S -p auto_start_on_new_cluster=false gzfsgc-rg
root@phys-newyork-1:~# cldg list dataz5
cldg: (C531091) Device group "dataz5" does not exist.
root@phys-newyork-1:~# zpool export dataz5
root@phys-newyork-1:~# clrs create -g gzfsgc-rg -t HAStoragePlus -p \
globalzpools=dataz5 hasp4dataz5
root@phys-newyork-1:~# cldg status dataz5

```

=== Cluster Device Groups ===

--- Device Group Status ---

Device Group Name	Primary	Secondary	Status
dataz5	-	-	Offline

```

root@phys-newyork-1:~# cldg show dataz5

```

=== Device Groups ===

```

Device Group Name:          dataz5
Type:                      ZPOOL
failback:                  false
Node List:                 phys-newyork-2, phys-newyork-1
preferenced:               false
autogen:                   true
numsecondaries:            1
ZFS pool name:             dataz5
poolaccess:                global

```

```

readonly:                false
import-at-boot:         false
searchpaths:             /dev/dsk

```

```

root@phys-newyork-1:~# clrg online -M gzfsqc-rg
root@phys-newyork-1:~# cldg status dataz5

```

```
=== Cluster Device Groups ===
```

```
--- Device Group Status ---
```

Device Group Name	Primary	Secondary	Status
dataz5	phys-newyork-2	phys-newyork-1	Online

```

root@phys-newyork-1:~# df -h | grep dataz5
dataz5                20G   42K    20G   1%   /dataz5
dataz5/fs1            20G   40K    20G   1%   /dataz5/fs1
root@phys-newyork-1:~#
root@phys-newyork-1:~# zpool list dataz5
cannot open 'dataz5': no such pool
root@phys-newyork-1:~#

```

The zpool is imported on one node of the global cluster but the filesystem is mounted on both global cluster nodes.

```

root@phys-newyork-2:~# df -h | grep dataz5
dataz5                20G   42K    20G   1%   /dataz5
dataz5/fs1            20G   40K    20G   1%   /dataz5/fs1
root@phys-newyork-2:~# zpool list dataz5
NAME    SIZE  ALLOC  FREE  CAP  DEDUP  HEALTH  ALTROOT
dataz5  19.9G  158K  19.9G   0%  1.00x  ONLINE  /
root@phys-newyork-2:~#

```

4. Configure the ZFS cluster file system in the zone cluster and create resource group and HAStoragePlus resource for it setting `rg_affinities` and `resource_dependencies_offline_restart` properties.

```

root@phys-newyork-1:~# clzc configure newyork-zc
clzc:newyork-zc> add fs
clzc:newyork-zc:fs> set dir=/global/apache
clzc:newyork-zc:fs> set special=/dataz5/fs1
clzc:newyork-zc:fs> set type=lofs
clzc:newyork-zc:fs> end

```



```

clzc:newyork-zc> exit
root@phys-newyork-1:~# clrg create -Z newyork-zc -S -p \
rg_affinities=++global:gzfgsc-rg -p auto_start_on_new_cluster=false apache-rg
root@phys-newyork-1:~# clr create -Z newyork-zc -t HASStoragePlus:12 -p \
filesystemmountpoints=/global/apache -p \
resource_dependencies_offline_restart=global:hasp4dataz5 -g apache-rg \
hasp4apache
root@phys-newyork-1:~# clrg online -M -Z newyork-zc apache-rg
root@phys-newyork-1:~# zlogin newyork-zc
[Connected to zone 'newyork-zc' pts/2]
Oracle Corporation      SunOS 5.11      st_015.server   February 2018
You have new mail.
root@newyork-zc-1:~# df -h | grep apache
/global/apache      20G   40K      20G    1%   /global/apache
root@newyork-zc-1:~#

```

The filesystem is mounted on both zone cluster nodes.

```

root@phys-newyork-2:~# zlogin newyork-zc
[Connected to zone 'newyork-zc' pts/4]
Oracle Corporation      SunOS 5.11      st_015.server   February 2018
You have new mail.
root@newyork-zc-2:~# df -h | grep apache
/global/apache      20G   40K      20G    1%   /global/apache
root@newyork-zc-2:~#

```

5. Configure apache in paris-zc.

```

root@phys-paris-1:~# zlogin paris-zc
[Connected to zone 'paris-zc' pts/3]
Oracle Corporation      SunOS 5.11      11.4    September 2018
You have new mail.
root@paris-zc-1:~# clrg create -p auto_start_on_new_cluster=false apachesip-rg
root@paris-zc-1:~# clr create -g apachesip-rg apache-lh
root@paris-zc-1:~# clrg set -p rg_dependencies=apachesip-rg apache-rg
root@paris-zc-1:~# cd /global/apache
root@paris-zc-1:/global/apache# df -k .
Filesystem      1024-blocks      Used  Available Capacity  Mounted on
/global/apache  15105524          42   15105482    1%   /global/apache
root@paris-zc-1:/global/apache# mkdir conf bin docs
root@paris-zc-1:/global/apache# cd /global/apache/conf
root@paris-zc-1:/global/apache/conf#
root@paris-zc-1:/global/apache/conf# cp /etc/apache2/2.4/httpd.conf .
<update httpd.conf>
root@paris-zc-1:/global/apache/conf# cd /global/apache/bin

```

```

root@paris-zc-1:/global/apache/bin# cp /usr/apache2/2.4/bin/apachectl .
root@paris-zc-1:/global/apache/bin#
<update apachectl>
root@paris-zc-1:/global/apache/bin# cd
root@paris-zc-1:~# clrg online -M apachesip-rg
root@paris-zc-1:~# clrt register apache
root@paris-zc-1:~# clrs create -g apache-rg -t apache -p bin_dir=/global/apache/bin \
-p resource_dependencies_offline_restart=hasp4apache -p \
resource_dependencies=apache-lh -p scalable=true apache

```

```

Oct 3 11:40:55 phys-paris-1 cl_runtime: NOTICE: Load balancer setting distribution
on paris-zc:apache:
Oct 3 11:40:55 phys-paris-1 cl_runtime: NOTICE: Node paris-zc-1: weight 1
Oct 3 11:40:55 phys-paris-1 cl_runtime: NOTICE: Node paris-zc-2: weight 1
Oct 3 11:40:55 phys-paris-1 cl_runtime: NOTICE: Load balancer setting distribution
on paris-zc:apache:
Oct 3 11:40:55 phys-paris-1 cl_runtime: NOTICE: Node paris-zc-1: weight 1
Oct 3 11:40:55 phys-paris-1 cl_runtime: NOTICE: Node paris-zc-2: weight 1
Oct 3 11:40:55 phys-paris-1 cl_runtime: NOTICE: Scalable service instance [TCP,
10.134.28.184,80] registered on node paris-zc-1.
Oct 3 11:41:01 phys-paris-1 cl_runtime: NOTICE: Scalable service instance [TCP,
10.134.28.184,80] registered on node paris-zc-2.
root@paris-zc-1:~# clrs status apache

```

=== Cluster Resources ===

Resource Name	Node Name	State	Status Message
apache	paris-zc-2	Online	Online - Completed successfully.
	paris-zc-1	Online	Online - Completed successfully.

```
root@paris-zc-1:~#
```

6. Set required permissions on the source and target filesystems.

- On phys-paris-2, where dataz4 zpool is imported:

```

root@phys-paris-2:~# zfs allow zfsuser1 create,destroy,\
hold,mount,receive,release,rollback,send,snapshot dataz4/fs1
root@phys-paris-2:~#

```

- On phys-newyork-2, where dataz5 zpool is imported:

```

root@phys-newyork-2:~# zfs allow zfsuser2 create,destroy,\
hold,mount,receive,release,rollback,send,snapshot dataz5/fs1
root@phys-newyork-2:~#

```

7. Update script-based plugins configuration file with the replication component for the ZFS snapshot protection group.

- On all nodes of paris:

```
root@phys-paris-1:~# echo "gzfs-fs1|any|phys-paris-1,phys-paris-2" \
>> /var/tmp/ody/sbp_conf
root@phys-paris-1:~#
```

- On all nodes of newyork:

```
root@phys-newyork-1:~# echo "gzfs-fs1|any|phys-newyork-1,phys-newyork-2" \
>> /var/tmp/ody/sbp_conf
root@phys-newyork-1:~#
```

8. From one node of paris create the ZFS snapshot protection group.

```
root@phys-paris-1:~# cat /var/tmp/ody/gzfs_gcp
PS=paris_newyork
PG=gzfspg
REPCOMP=gzfs-fs1
REPRS=gzfs-fs1-rep
REPRG=gzfs-fs1-rep-rg
DESC="GZFS with zfs snapshot replication"
APPRG=gzfs_gc-rg
CONFIGFILE=/var/tmp/ody/sbp_conf
LOCAL_REP_USER=zfsuser1
REMOTE_REP_USER=zfsuser2
LOCAL_PRIV_KEY_FILE=/export/home/zfsuser1/.ssh/id_rsa
REMOTE_PRIV_KEY_FILE=/export/home/zfsuser2/.ssh/id_rsa
LOCAL_ZPOOL_RS=hasp4dataz4
REMOTE_ZPOOL_RS=hasp4dataz5
LOCAL_LH=paris-rep-lh
REMOTE_LH=newyork-rep-lh
LOCAL_DATASET=dataz4/fs1
REMOTE_DATASET=dataz5/fs1
REPLICATION_INTERVAL=
NUM_OF_SNAPSHOTS_TO_STORE=
REPLICATION_STREAM_PACKAGE=
SEND_PROPERTIES=
INTERMEDIARY_SNAPSHOTS=
RECURSIVE=
MODIFY_PASSPHRASE=
root@phys-paris-1:~# clrs disable -g gzfs_gc-rg +
Oct  3 11:55:18 phys-paris-1 cl_runtime: NOTICE: Scalable service instance [TCP,
10.134.28.184,80] deregistered on node paris-zc-2.
```

```
root@phys-paris-1:~# Oct 3 11:55:20 phys-paris-1 cl_runtime: NOTICE: Scalable
service instance [TCP,10.134.28.184,80] deregistered on node paris-zc-1.
```

```
root@phys-paris-1:~# clrg offline gzfs-gc-rg
root@phys-paris-1:~# clrg unmanage gzfs-gc-rg
root@phys-paris-1:~# clrg status -Z paris-zc apache-rg
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
paris-zc:apache-rg	paris-zc-2	No	Offline
	paris-zc-1	No	Offline

```
root@phys-paris-1:~# geops status paris_newyork
```

```
Partnership "paris_newyork"      : OK
Partner clusters                 : newyork
Synchronization                  : OK
ICRM Connection                  : OK
```

```
Heartbeat "hb_paris~newyork" monitoring "newyork": OK
Plug-in "ping_plugin"           : Inactive
Plug-in "tcp_udp_plugin"        : OK
```

```
root@phys-paris-1:~# /opt/ORCLscgprepzfssnap/util/zfs_snap_geo_register \
-f /var/tmp/ody/gzfs-gcpg
```

```
sourcing /var/tmp/ody/gzfs-gcpg
```

```
Creating protection group "gzfs-gcpg"
This operation might take a while...
Protection group "gzfs-gcpg" successfully created.
```

```
Adding the replication component "gzfs-fs1" to the protection group "gzfs-gcpg"
This operation might take a while...
Password for property local_service_password :
Password for property remote_service_password :
Script-based plug-in configuration "gzfs-fs1" successfully added to the protection
group "gzfs-gcpg".
```

```
Wait for 60 seconds to allow potential locks to be released
```

```
Adding the application resource group(s) "gzfsgc-rg" to "gzfsgcpg"
Following resource groups successfully added:
    "gzfsgc-rg".
```

Run "geopg get --partnership paris_newyork gzfsgcpg" on the partner cluster if this has not been done.

```
root@phys-paris-1:~#
```

9. Get the protection group from one node of newyork global cluster.

```
root@phys-newyork-1:~# clrs disable -g gzfsgc-rg +
root@phys-newyork-1:~# clrg offline gzfsgc-rg
root@phys-newyork-1:~# clrg unmanage gzfsgc-rg
root@phys-newyork-1:~# clrg status -Z newyork-zc apache-rg
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
newyork-zc:apache-rg	newyork-zc-2	No	Offline
	newyork-zc-1	No	Offline

```
root@phys-newyork-1:~# geopg get --partnership paris_newyork gzfsgcpg
Protection group "gzfsgcpg" successfully created.
root@phys-newyork-1:~#
```

10. From one node of paris-zc create protection group of type NONE and add application resource groups to it.

```
root@phys-paris-1:~# zlogin paris-zc
[Connected to zone 'paris-zc' pts/3]
Oracle Corporation    SunOS 5.11    11.4    September 2018
You have new mail.
root@paris-zc-1:~# geops status paris-zc_newyork-zc
```

```
Partnership "paris-zc_newyork-zc"      : OK
Partner clusters      : newyork-zc
Synchronization      : OK
ICRM Connection      : OK
```

```
Heartbeat "hb_paris-zc~newyork-zc" monitoring "newyork-zc": OK
Plug-in "ping_plugin"      : Inactive
Plug-in "tcp_udp_plugin"   : OK
```

```
root@paris-zc-1:~# geopg create -s paris-zc_newyork-zc -o primary -d \
```

```

none -p External_Dependency_Allowed=true gzfszcp
Protection group "gzfszcp" successfully created.
root@paris-zc-1:~# clrg status apachesip-rg apache-rg

```

```

=== Cluster Resource Groups ===

```

Group Name	Node Name	Suspended	Status
apachesip-rg	paris-zc-2	No	Online
	paris-zc-1	No	Offline
apache-rg	paris-zc-2	No	Offline
	paris-zc-1	No	Offline

```

root@paris-zc-1:~# clrs disable -g apachesip-rg,apache-rg +
root@paris-zc-1:~# clrg unmanage apachesip-rg apache-rg
clrg: (C605916) Cannot make resource group apachesip-rg UNMANAGED because it is in a
non-OFFLINE state on one or more nodes in the cluster
root@paris-zc-1:~# Oct 3 12:17:10 phys-paris-1 cl_runtime: NOTICE: load balancer for
group 'paris-zc:apache' released

```

```

root@paris-zc-1:~# clrg offline apachesip-rg apache-rg
clrg: (C328207) Skipping resource group apache-rg because it is unmanaged
root@paris-zc-1:~# Oct 3 12:17:30 phys-paris-1 cl_runtime: NOTICE: load balancer
deleted

```

```

root@paris-zc-1:~# clrg unmanage apachesip-rg apache-rg
root@paris-zc-1:~# geopg add-resource-group apachesip-rg,apache-rg gzfszcp
Following resource groups successfully added:
"apachesip-rg,apache-rg".
root@paris-zc-1:~#

```

11. From one node of newyork-zc, get the protection group from paris-zc.

```

root@phys-newyork-1:~# zlogin newyork-zc
[Connected to zone 'newyork-zc' pts/2]
Oracle Corporation      SunOS 5.11      st_015.server   February 2018
You have new mail.
root@newyork-zc-1:~# clrg create -p auto_start_on_new_cluster=false apachesip-rg
root@newyork-zc-1:~# clrs disable -g apachesip-rg,apache-rg +
root@newyork-zc-1:~# clrg offline apachesip-rg apache-rg
clrg: (C328207) Skipping resource group apachesip-rg because it is unmanaged
root@newyork-zc-1:~# clrg unmanage apachesip-rg apache-rg
root@newyork-zc-1:~# geopg get -s paris-zc_newyork-zc gzfszcp

```

```
Protection group "gzfszcpq" successfully created.
root@newyork-zc-1:~#
```

12. On the paris site, create site, join site, create multigroup, and add protection groups to the multigroup setting dependency from the zone cluster protection group to the global cluster protection group.

```
root@phys-paris-1:~# geosite create -m paris-zc paris-site
Processing operation...might take several minutes to complete.
Site "paris-site" successfully created.
root@phys-paris-1:~# geomg create -s paris-site parismg
Processing operation...might take several minutes to complete.
```

```
Multigroup "parismg" successfully created.
root@phys-paris-1:~# zlogin paris-zc
[Connected to zone 'paris-zc' pts/3]
Oracle Corporation      SunOS 5.11      11.4      September 2018
You have new mail.
root@paris-zc-1:~# geosite join paris paris-site
Processing operation...might take several minutes to complete.
Local cluster "paris-zc" successfully joined site "paris-site" on cluster "paris".
root@paris-zc-1:~# exit
logout
```

```
[Connection to zone 'paris-zc' pts/3 closed]
root@phys-paris-1:~# geomg add-protection-group \
paris-zc:gzfszcpq/paris:gzfsgcpq parismg
The following protection group list was successfully added to multigroup "parismg":
    "paris-zc:gzfszcpq/paris:gzfsgcpq".
root@phys-paris-1:~# geomg status -v parismg
```

Multigroup: parismg

```
Description          :
Site name             : paris-site
Creation signature    : paris Oct 3, 2018 12:28:11 PM PDT
Last update           : Oct 3, 2018 12:30:19 PM PDT
Configuration status  : OK

Synchronization status with :
    paris-zc          : OK

Timeout               : 7320 seconds
Protection group list  : paris-zc:gzfszcpq/paris:gzfsgcpq
```

```
Status of protection group "paris:gzfsgcpg"      :
Global status      : Unknown
Partnership        : paris_newyork
Synchronization    : OK

Cluster paris      : Unknown
Role               : Primary
Activation state    : Deactivated
Configuration      : OK
Data replication    : Unknown
Resource groups    : OK

Cluster newyork    : Unknown
Role               : Secondary
Activation state    : Deactivated
Configuration      : OK
Data replication    : Unknown
Resource groups    : OK

Status of protection group "paris-zc:gzfszcp" :
Global status      : OK
Partnership        : paris-zc_newyork-zc
Synchronization    : OK

Cluster paris-zc   : OK
Role               : Primary
Activation state    : Deactivated
Configuration      : OK
Data replication    : None
Resource groups    : OK

Cluster newyork-zc : OK
Role               : Secondary
Activation state    : Deactivated
Configuration      : OK
Data replication    : None
Resource groups    : OK
```

```
root@phys-paris-1:~#
```

13. On newyork site, create site, join site, create multigroup, and add protection groups to the multigroup setting dependency from the zone cluster protection group to the global cluster protection group.


```

root@phys-newyork-1:~# geosite create -m newyork-zc newyork-site
Processing operation...might take several minutes to complete.
Site "newyork-site" successfully created.
root@phys-newyork-1:~# geomg create -s newyork-site newyorkmg
Processing operation...might take several minutes to complete.

Multigroup "newyorkmg" successfully created.
root@phys-newyork-1:~# zlogin newyork-zc
[Connected to zone 'newyork-zc' pts/2]
Oracle Corporation      SunOS 5.11      st_015.server   February 2018
You have new mail.
root@newyork-zc-1:~# geosite join newyork newyork-site
Processing operation...might take several minutes to complete.
Local cluster "newyork-zc" successfully joined site "newyork-site" on cluster
"newyork".
root@newyork-zc-1:~# exit
logout

[Connection to zone 'newyork-zc' pts/2 closed]
root@phys-newyork-1:~# geomg add-protection-group \
newyork-zc:gzfszcpng/newyork:gzfsgcpg newyorkmg
The following protection group list was successfully added to multigroup
"newyorkmg":
    "newyork-zc:gzfszcpng/newyork:gzfsgcpg".
root@phys-newyork-1:~# geomg status -v newyorkmg

Multigroup: newyorkmg

Description          :
Site name            : newyork-site
Creation signature    : newyork Oct 3, 2018 12:34:08 PM PDT
Last update          : Oct 3, 2018 12:36:09 PM PDT
Configuration status  : OK

Synchronization status with :
    newyork-zc          : OK

Timeout              : 7320 seconds
Protection group list  : newyork-zc:gzfszcpng/newyork:gzfsgcpg

Status of protection group "newyork-zc:gzfszcpng" :
    Global status      : OK
    Partnership        : paris-zc_newyork-zc

```

```
Synchronization          : OK

Cluster newyork-zc      : OK
  Role                   : Secondary
  Activation state      : Deactivated
  Configuration         : OK
  Data replication     : None
  Resource groups      : OK

Cluster paris-zc       : OK
  Role                   : Primary
  Activation state      : Deactivated
  Configuration         : OK
  Data replication     : None
  Resource groups      : OK

Status of protection group "newyork:gzfsgcpg" :
  Global status         : Unknown
  Partnership           : paris_newyork
  Synchronization      : OK

Cluster newyork        : Unknown
  Role                   : Secondary
  Activation state      : Deactivated
  Configuration         : OK
  Data replication     : Unknown
  Resource groups      : OK

Cluster paris          : Unknown
  Role                   : Primary
  Activation state      : Deactivated
  Configuration         : OK
  Data replication     : Unknown
  Resource groups      : OK
```

```
root@phys-newyork-1:~#
```

14. Start the protection group using the `geomg` command.

```
root@phys-paris-1:~# geomg start -e global parismg
Processing operation... The timeout period for this multigroup operation is 7320
seconds...
```

```
Beginning "Start" operation on protection group "paris:gzfsgcpg"
```

```

Oct  3 12:41:05 phys-paris-1 cl_runtime: NOTICE: Load balancer setting distribution
on paris-zc:apache:
Oct  3 12:41:05 phys-paris-1 cl_runtime: NOTICE: Node paris-zc-1: weight 1
Oct  3 12:41:05 phys-paris-1 cl_runtime: NOTICE: Node paris-zc-2: weight 1
Oct  3 12:41:05 phys-paris-1 cl_runtime: NOTICE: Scalable service instance [TCP,
10.134.28.184,80] registered on node paris-zc-1.
"Start" operation on protection group "paris:gzfsgcpg" completed successfully
Beginning "Start" operation on protection group "paris-zc:gzfszcp"
Oct  3 12:41:10 phys-paris-1 cl_runtime: NOTICE: Scalable service instance [TCP,
10.134.28.184,80] registered on node paris-zc-2.
"Start" operation on protection group "paris-zc:gzfszcp" completed successfully

```

```

Multigroup "parismg" successfully started.
root@phys-paris-1:~# geomg status -v parismg

```

Multigroup: parismg

```

Description          :
Site name            : paris-site
Creation signature   : paris Oct 3, 2018 12:28:11 PM PDT
Last update          : Oct 3, 2018 12:30:19 PM PDT
Configuration status : OK

Synchronization status with :
    paris-zc          : OK

Timeout              : 7320 seconds
Protection group list : paris-zc:gzfszcp/paris:gzfsgcpg

Status of protection group "paris:gzfsgcpg" :
    Global status     : OK
    Partnership       : paris_newyork
    Synchronization   : OK

    Cluster paris     : OK
        Role          : Primary
        Activation state : Activated
        Configuration   : OK
        Data replication : OK
        Resource groups : OK

    Cluster newyork   : OK
        Role          : Secondary

```

```

Activation state : Activated
Configuration    : OK
Data replication : OK
Resource groups  : OK

```

```

Status of protection group "paris-zc:gzfszcp" :
Global status      : OK
Partnership        : paris-zc_newyork-zc
Synchronization   : OK

```

```

Cluster paris-zc      : OK
Role                  : Primary
Activation state      : Activated
Configuration         : OK
Data replication      : None
Resource groups       : OK

```

```

Cluster newyork-zc   : OK
Role                  : Secondary
Activation state      : Activated
Configuration         : OK
Data replication      : None
Resource groups       : OK

```

```
root@phys-paris-1:~# clrs status -Z paris-zc apache
```

```
=== Cluster Resources ===
```

Resource Name	Node Name	State	Status Message
apache	paris-zc-2	Online	Online - Service is online.
	paris-zc-1	Online	Online - Service is online.

```
root@phys-paris-1:~#
```

15. In the Zone cluster, add data to the ZFS-based cluster filesystem that is being replicated. This is for illustrating replication functioning after switchover in later step.

```

root@phys-paris-1:~# zlogin paris-zc
[Connected to zone 'paris-zc' pts/3]
Oracle Corporation   SunOS 5.11      11.4   September 2018
You have new mail.
root@paris-zc-1:~# mkdir /global/apache/docs/testdir
root@paris-zc-1:~# cp -r /net/howtos/QA_Docs/* /global/apache/docs/testdir

```

```
root@paris-zc-1:~# ls /global/apache/docs/testdir
BestPractice-pplsft.odt
howto-2.sxw
howto-Template.odt
HA-Oracle-on-ARNAS
Oracle.Solaris.Cluster.Cheat.Sheet.Template.9.16.10.odt
PeopleSoft_and_SolarisCluster_install_and_config_procedure.odt
PeopleSoft_and_SolarisCluster_install_and_config_procedure.odt.copy
Peoplesoft_scheduler_setup_instructions.txt
Peoplesoft_scheduler_setup_instructions.txt.old
README--Install-GIB3-in-zone-cluster-v1.odt
Solaris_Cluster_3.3_RAC_10g_LDOM.docx
Solaris_Cluster_3.3_RAC_10g_LDOM.odt
Solaris_Cluster_3.3_RAC_10g_LDOM.pdf
Technical.Instruction.Template.ott
cheatsheet-HA-PeopleSoft-zone-cluster.txt
cheatsheet-setup-HA-PeopleSoft-process-scheduler-ZC.txt
cluster-webas-zones-install.pdf
geo2sitesHTG.odt
how_to_run_scate.odt
how_to_run_scate.txt
jumbo.odt
mpxio.odt
oldcli_to_newcli_mapping.odt
oldcli_to_newcli_mapping.odt.copy
problems_solutions.txt
quorum_server_setup.txt
rac_upgrade.odt
rac_upgrade.txt
scate_ha_oracle_load.odt
scate_rac_load.odt
scate_requirements.html
scge-odg-ooow.odt
setup_avs.odt
setup_avs.txt
setup_ipv6.odt
setup_ipv6.txt
setup_ipv6.txt.copy
setup_samp_2zc.txt
setup_tx.odt
setup_tx.txt
vxvm_document.odt
xxPeoplesoft_scheduler_setup_instructions.txtxx
```

```
root@paris-zc-1:~#
```

16. Switchover multigroup to newyork site, verify that data in the filesystem exists on this cluster, and complete setting up resources for apache on this site.

```
root@phys-newyork-1:~# geomg status -v newyorkmg
```

```
Multigroup: newyorkmg
```

```
Description          :
Site name            : newyork-site
Creation signature   : newyork Oct 3, 2018 12:34:08 PM PDT
Last update         : Oct 3, 2018 12:36:09 PM PDT
Configuration status : OK

Synchronization status with :
                        newyork-zc          : OK

Timeout              : 7320 seconds
Protection group list : newyork-zc:gzfsczpcg/newyork:gzfsgcpg
```

```
Status of protection group "newyork-zc:gzfsczpcg" :
Global status      : OK
Partnership        : paris-zc_newyork-zc
Synchronization   : OK
```

```
Cluster newyork-zc          : OK
Role                      : Secondary
Activation state           : Activated
Configuration              : OK
Data replication           : None
Resource groups            : OK
```

```
Cluster paris-zc          : OK
Role                      : Primary
Activation state           : Activated
Configuration              : OK
Data replication           : None
Resource groups            : OK
```

```
Status of protection group "newyork:gzfsgcpg" :
Global status      : OK
Partnership        : paris_newyork
Synchronization   : OK
```

```

Cluster newyork                : OK
  Role                          : Secondary
  Activation state               : Activated
  Configuration                  : OK
  Data replication               : OK
  Resource groups                : OK

```

```

Cluster paris                   : OK
  Role                          : Primary
  Activation state               : Activated
  Configuration                  : OK
  Data replication               : OK
  Resource groups                : OK

```

```

root@phys-newyork-1:~# geomg switchover -s newyork-site newyorkmg
Are you sure you want to switch over multigroup 'newyorkmg' to site 'newyork-site'?
(yes|no) > yes
Processing operation... The timeout period for this multigroup operation is 7320
seconds...

```

```

Beginning "Switchover" operation on protection group "newyork-zc:gzfsczpcpg"
Beginning "Switchover" operation on protection group "newyork:gzfsgcpg"
"Switchover" operation on protection group "newyork:gzfsgcpg" completed successfully
"Switchover" operation on protection group "newyork-zc:gzfsczpcpg" completed
successfully

```

```

"Switchover" operation succeeded for the multigroup "newyorkmg".

```

```

root@phys-newyork-1:~# geomg status -v newyorkmg

```

```

Multigroup: newyorkmg

```

```

Description                    :
Site name                       : newyork-site
Creation signature              : newyork Oct 3, 2018 12:34:08 PM PDT
Last update                     : Oct 3, 2018 12:36:09 PM PDT
Configuration status            : OK

```

```

Synchronization status with    :
  newyork-zc                    : OK

```

```

Timeout                         : 7320 seconds
Protection group list           : newyork-zc:gzfsczpcpg/newyork:gzfsgcpg

```

```
Status of protection group "newyork-zc:gzfsczpcpg"      :
  Global status      : OK
  Partnership        : paris-zc_newyork-zc
  Synchronization    : OK

Cluster newyork-zc      : OK
  Role                : Primary
  Activation state     : Activated
  Configuration        : OK
  Data replication     : None
  Resource groups      : OK

Cluster paris-zc       : OK
  Role                : Secondary
  Activation state     : Activated
  Configuration        : OK
  Data replication     : None
  Resource groups      : OK

Status of protection group "newyork:gzfsgcpg"          :
  Global status      : OK
  Partnership        : paris_newyork
  Synchronization    : OK

Cluster newyork        : OK
  Role                : Primary
  Activation state     : Activated
  Configuration        : OK
  Data replication     : OK
  Resource groups      : OK

Cluster paris          : OK
  Role                : Secondary
  Activation state     : Activated
  Configuration        : OK
  Data replication     : OK
  Resource groups      : OK

root@phys-newyork-1:~# zlogin newyork-zc
[Connected to zone 'newyork-zc' pts/2]
Oracle Corporation      SunOS 5.11      st_015.server  February 2018
You have new mail.
```



```
root@newyork-zc-1:~# ls /global/apache/docs/testdir
BestPractice-pplsft.odt
howto-2.sxw
howto-Template.odt
HA-Oracle-on-ARNAS
Oracle.Solaris.Cluster.Cheat.Sheet.Template.9.16.10.odt
PeopleSoft_and_SolarisCluster_install_and_config_procedure.odt
PeopleSoft_and_SolarisCluster_install_and_config_procedure.odt.copy
Peoplesoft_scheduler_setup_instructions.txt
Peoplesoft_scheduler_setup_instructions.txt.old
README--Install-GIB3-in-zone-cluster-v1.odt
Solaris_Cluster_3.3_RAC_10g_LDOM.docx
Solaris_Cluster_3.3_RAC_10g_LDOM.odt
Solaris_Cluster_3.3_RAC_10g_LDOM.pdf
Technical.Instruction.Template.ott
cheatsheet-HA-PeopleSoft-zone-cluster.txt
cheatsheet-setup-HA-PeopleSoft-process-scheduler-ZC.txt
cluster-webas-zones-install.pdf
geo2sitesHTG.odt
how_to_run_scate.odt
how_to_run_scate.txt
jumbo.odt
mpxio.odt
oldcli_to_newcli_mapping.odt
oldcli_to_newcli_mapping.odt.copy
problems_solutions.txt
quorum_server_setup.txt
rac_upgrade.odt
rac_upgrade.txt
scate_ha_oracle_load.odt
scate_rac_load.odt
scate_requirements.html
scge-odg-ooow.odt
setup_avs.odt
setup_avs.txt
setup_ipv6.odt
setup_ipv6.txt
setup_ipv6.txt.copy
setup_samp_2zc.txt
setup_tx.odt
setup_tx.txt
vxvm_document.odt
xxPeoplesoft_scheduler_setup_instructions.txtxx
```

```

root@newyork-zc-1:~# clrg status apachesip-rg apache-rg

=== Cluster Resource Groups ===

Group Name      Node Name      Suspended      Status
-----
apachesip-rg    newyork-zc-2   No             Online
                 newyork-zc-1   No             Offline

apache-rg       newyork-zc-2   No             Online
                 newyork-zc-1   No             Online

root@newyork-zc-1:~# clrs list -g apachesip-rg
root@newyork-zc-1:~# clrs list -g apache-rg hasp4apache
root@newyork-zc-1:~# clrssa create -g apachesip-rg apache-lh
root@newyork-zc-1:~# clrg set -p rg_dependencies=apachesip-rg apache-rg
root@newyork-zc-1:~# clrt register apache
root@newyork-zc-1:~# clrs create -g apache-rg -t apache -p bin_dir=/global/apache/bin
\
-p resource_dependencies_offline_restart=hasp4apache -p \
resource_dependencies=apache-lh -p scalable=true apache
Oct 3 14:33:16 phys-newyork-1 cl_runtime: NOTICE: Service group 'newyork-zc:apache'
created
Oct 3 14:33:16 phys-newyork-1 cl_runtime: NOTICE: Service object [TCP,
10.134.28.184, 80] created in group 'newyork-zc:apache'
Oct 3 14:33:16 phys-newyork-1 cl_runtime: NOTICE: Load balancer setting distribution
on newyork-zc:apache:
Oct 3 14:33:16 phys-newyork-1 cl_runtime: NOTICE: Node newyork-zc-1: weight 1
Oct 3 14:33:16 phys-newyork-1 cl_runtime: NOTICE: Node newyork-zc-2: weight 1
Oct 3 14:33:17 phys-newyork-1 cl_runtime: NOTICE: Scalable service instance [TCP,
10.134.28.184,80] registered on node newyork-zc-1.
Oct 3 14:33:17 phys-newyork-1 cl_runtime: NOTICE: Scalable service instance [TCP,
10.134.28.184,80] registered on node newyork-zc-2.
root@newyork-zc-1:~# clrs status apache

=== Cluster Resources ===

Resource Name    Node Name      State      Status Message
-----
apache           newyork-zc-2   Online     Online - Completed successfully.
                 newyork-zc-1   Online     Online - Completed successfully.

root@newyork-zc-1:~# geomg status -v newyorkmg

```

Multigroup: newyorkmg

Description :
Site name : newyork-site
Creation signature : newyork Oct 3, 2018 12:34:08 PM PDT
Last update : Oct 3, 2018 12:36:09 PM PDT
Configuration status : OK

Synchronization status with :
newyork : OK

Timeout : 7320 seconds
Protection group list : newyork-zc:gzfszcpq/newyork:gzfsgcpg

Status of protection group "newyork-zc:gzfszcpq" :
Global status : OK
Partnership : paris-zc_newyork-zc
Synchronization : OK

Cluster newyork-zc : OK
Role : Primary
Activation state : Activated
Configuration : OK
Data replication : None
Resource groups : OK

Cluster paris-zc : OK
Role : Secondary
Activation state : Activated
Configuration : OK
Data replication : None
Resource groups : OK

Status of protection group "newyork:gzfsgcpg" :
Global status : OK
Partnership : paris_newyork
Synchronization : OK

Cluster newyork : OK
Role : Primary
Activation state : Activated
Configuration : OK

```
Data replication : OK  
Resource groups : OK
```

```
Cluster paris      : OK  
Role              : Secondary  
Activation state  : Activated  
Configuration     : OK  
Data replication  : OK  
Resource groups   : OK
```

```
root@newyork-zc-1:~#
```

Index

A

adding

- application resource groups
 - troubleshooting, 53
- replication components
 - troubleshooting, 49

administering

- remote replication with, 77

application resource groups

- administering, 67
- changing resource group type, 66
- creating, 50
- removing, 67

C

configuration file, 26

configuring

- protection groups
 - instructions, 34
 - troubleshooting, 40

creating

- application resource group, 50
- protection groups
 - instructions, 34

D

data recovery, 80

- failback-switchover, 81
- failback-takeover, 84
- switchover, 77

data replication using Oracle Solaris ZFS snapshot

overview, 11

DEBUG property, 40

debugging protection groups, 40

deleting

- application resource group, 67

E

extension properties

- replication agent resource, 91
- resource status, 95

F

failback-switchover, 81

failback-takeover, 84

G

guidelines, 25

I

incremental replication, 18

infrastructure resource groups, 19

intermediary snapshots, 18

O

/opt/ORCLscgprepzfssa/etc/zfssa_geo_config file, 61

/opt/ORCLscgprepzfssnap/util/

zfs_snap_geo_register script, 47, 61

- Oracle Solaris ZFS snapshot
 - administering remote replication with, 77
 - application resource groups
 - adding to protection group, 50
 - Removing, 67
 - removing, 67
 - configuration file, 61
 - data recovery
 - failback-switchover, 77, 81
 - failback-takeover, 84
 - protection groups
 - creating, 34
 - recovering from errors, 87
 - registration script, 47, 61
 - runtime status, 72
 - detailed, 74
 - overall, 72
 - state and status messages, 74
 - zfs_snap_geo_register script, 47
 - zfsa_geo_config file, 61
 - zfsa_geo_register script, 61
 - Oracle ZFS Storage
 - protection groups
 - replicating configuration of, 56
- P**
- planning, 25
 - prerequisites for configuring replication, 32
 - primary cluster
 - data recovery, 80
 - properties
 - ZFS snapshots, 91
 - protection group
 - removing
 - troubleshooting, 69
 - protection groups
 - adding application resource group to, 50
 - adding application resource groups
 - troubleshooting, 53
 - adding replication components
 - troubleshooting, 49
 - configuring, 34
 - troubleshooting, 40
 - creating, 34
 - debugging, 40
 - removing application resource group, 67
 - replicating configuration of, 56
 - starting, 42
 - stopping, 43
 - validating
 - troubleshooting, 59
- R**
- recovery *See* data recovery
 - from replication error, 87
 - recursive stream package, 17
 - removing
 - protection group
 - troubleshooting, 69
 - replication
 - Oracle Solaris ZFS snapshot, 77
 - protection group configuration, 56
 - recovering from errors, 87
 - runtime status details, 74
 - runtime status overview, 72
 - replication agent resource groups, 21
 - replication configuration file, 26
 - replication status resource groups, 23
 - replication stream package, 16
 - resource groups
 - application, 67
 - changing type, 66
 - infrastructure, 19
 - replication agent, 21
 - replication status, 23, 74
 - resource log file, 41
 - restrictions, 25
 - resumable replication, 15
 - runtime status
 - replication, 72
 - state and status messages, 74

S

- starting protection groups, 42
- stopping protection groups, 43
- switchover, 77

T

- takeover
 - failback-switchover, 81
 - failback-takeover, 84
 - switchover, 77
- trace log file, 41
- troubleshooting
 - adding application resource groups to protection groups, 53
 - adding remote replication components to a protection group, 49
 - configuring protection groups, 40
 - debugging protection groups, 40
 - removing a protection group, 69
 - validating protection groups, 59

V

- validating
 - protection groups
 - troubleshooting, 59

Z

- ZFS snapshots
 - properties of, 91
- ZFS snapshots configuration file, 61
- ZFS snapshots registration script, 47, 61
- zfs_snap_geo_config file, 61
- zfs_snap_geo_register script, 47, 61

