

# Seeding from a DR Series System to an External Device

## Technical White Paper

Quest Engineering

December 2017



## Copyright 2018 Quest Software Inc.

### ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.

Attn: LEGAL Dept

4 Polaris Way

Aliso Viejo, CA 92656

Refer to our Web site ([www.quest.com](http://www.quest.com)) for regional and international office information.


### Patents


Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at [www.quest.com/legal](http://www.quest.com/legal).

### Trademarks

Quest, and the Quest logo are trademarks and registered trademarks of Quest Software Inc. in the U.S.A. and other countries. For a complete list of Quest Software trademarks, please visit our website at [www.quest.com/legal](http://www.quest.com/legal). All other trademarks, servicemarks, registered trademarks, and registered servicemarks are the property of their respective owners.

### Legend

 **WARNING:** A WARNING icon indicates a potential for property damage, personal injury, or death

 **CAUTION:** A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

 **IMPORTANT, NOTE, TIP, MOBILE, or VIDEO:** An information icon indicates supporting information.

# Contents

<b>Executive Summary</b> .....	<b>4</b>
<b>Understanding the seeding process</b> .....	<b>5</b>
<b>Seeding data from a DRSeries system and copying it to a replication target</b> .....	<b>7</b>
Preparing for seeding.....	7
Setting up a seeding export job on the source DRSeries system.....	7
Important notes about the seeding export process.....	10
Export CLI flow.....	11
<b>Setting up a seeding import job on the target DRSeries system</b> .....	<b>12</b>
Important notes about the import seeding process .....	15
Import CLI flow .....	16
<b>Synchronizing the source and target containers</b> .....	<b>17</b>
If a target container exists.....	17
If the target container does not exist.....	17

# Executive summary

---

This document provides information about how to set up seeding from a DR Series system to an external device by using the DR Series command line interface (CLI). This document is a quick reference guide and does not include all DR Series system deployment best practices.

For additional information, see the DR Series system documentation and other data management application best practices whitepapers for your specific DR Series system at:

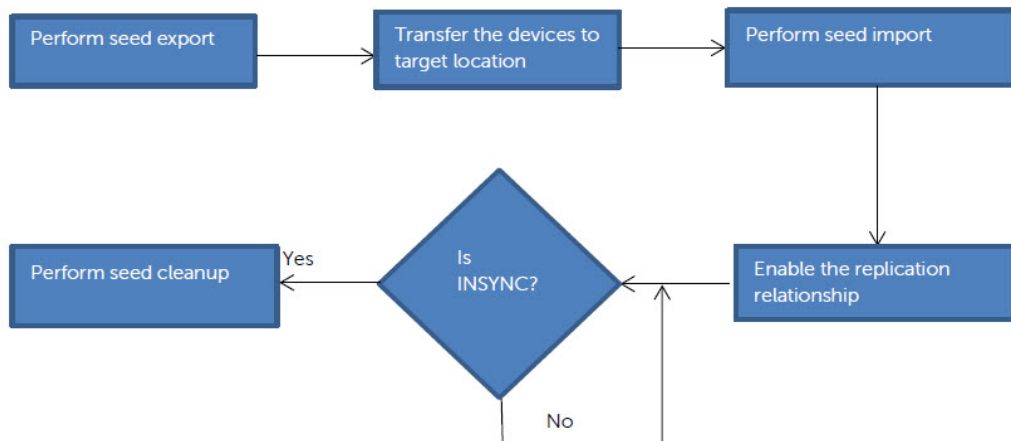
<http://support.quest.com/DR-Series>

## Understanding the seeding process

Seeding is a process that copies de-duplicated data from a DR Series system to an external device (typically USB), which is exported as a CIFS share. The seeding process comprises the following steps:

- 1 The device attached to a client machine is mounted by using the CIFS protocol on the DR Series system; and, seeding is initiated, which copies the data (export) to the device.
- 2 Once the data is copied entirely, the devices are later shipped to the location of the target DR Series system.
- 3 The device is attached to another client at the remote site and mounted by using the CIFS protocol on the target DR Series system.
- 4 The data is imported to the target DR Series system.
- 5 Once all of the data is imported, replication re-sync is initiated between the source and target DR containers. Since the data is already present on the target, the re-sync completes quickly after transferring the namespace with any other changes in data. Replication setup is ready between the source and target.

The following diagram shows this process.



---

# Seeding data from a DR Series system and copying it to a replication target

## Preparing for seeding

Before the seeding process is started, you need to gather the following information.

- The names of the containers on the appliance that contain the data to be replicated (typically the containers for which the disaster recovery setup must be done). These container names are given as input to the seeding process.
- Name or IP address of the client machine on which the USB device is mounted.
- Name of the CIFS share (USB device) exported to the “outside world.”
- Credentials (including domain, if any) to access the client.
- A password (minimum eight characters) to use for encrypting the data on the device. This password must be entered at import time to decrypt the data.

## Setting up a seeding export job on the source DR Series system

The following steps describe the commands for setting up a seeding export job on the source DR Series system.

- 1 Create a seed export job by entering the following command.

```
seed --create --op export [--enc_type <aes128 | aes256>]
```

By default the data is encrypted; however, the encryption type is optional. This command would request a password that is needed for encryption as shown in the following example. (Note that this same password is to be given as input to the import operation, which verifies and decrypts the data.)

```
[root@DR4000-DHYLBM1 ~]# seed --create --op export --enc_type aes128
Enter password for seed export:
Re-enter password for seed export:
Successfully created seed job details.
```

- 2 Once a seed job is created, the container whose data is to be seeded must be added to the job. If there are multiple containers, you can execute the following command multiple times to seed the data from all containers.

```
seed --add_container --name <container name>
```

```
[root@DR4000-DHYLBM1 ~]# seed --add_container --name seed_container
Successfully added seed container seed_container.
```

- 3 To reference the device to which the seed data must be copied, enter the following command. (Note that only shares exported by CIFS protocol are recognized.)

```
seed --add_device --server <server name> --volume <volume> --
username
<user name> [--domain <domain name>]
```

In this command, the parameters are as follows:

- server name - the name of the client which exports the device as a CIFS share.
- volume name - the name of the export share on the target server. The name is case-sensitive.
- user name - credentials to access the share. Password is requested separately.

```
[root@DR4000-DHYLBM1 ~]# seed --add_device --server 10.250.242.213 --volume win_share --username administrator
Enter password for administrator@10.250.242.213:
Successfully added seed device.
```

- 4 Once the add device is successful, you can cross check the configuration details by using the following command. This step is recommended before you proceed further.

```
seed -show
```

```
[root@DR4000-DHYLBM1 ~]# seed --show
Device info
=====
Server                :10.250.242.213
Volume                :win_share
Username              :administrator
Domain                :NA

Job info
=====
Operation             :Export
Status                :Initialized
Container             :seed_container
Encryption type       :aes128
```



- 5 Enter the following command to start the seeding process.

### **seed -start**

The data copy begins and continues until either the space on the device is full or all of the data of the containers added is seeded.

```
[root@DR4000-DHYLBM1 ~]# seed --start  
Successfully started seed job.
```

- 6 Enter the following command to monitor the seeding progress. If the "Seed status" has reached 'FINISHED', then the seeding of the containers given as input is complete.

### **seed -stats**

If the status is shown as 'Paused (Device full),' then the share has no more space to copy the data. The device must be removed (using the commands, "seed -stop" and "seed -remove\_device") and a new device with free space can be added (using the command, seed - add\_device). This scenario occurs when more devices are available, and seeding has to be continued. Adding a new device without deleting the job would continue to copy the data from where it was left. You can execute the command, "seed -start" to continue.

The type of information in the output of the command, "seed -stats," looks similar to the following example.

```
[root@DR4000-DHYLBM1 ~]# seed --stats  
Seeding Source Stats:  
  
Seed state:                SEED_STARTED  
Seed status:               PENDING  
Seed device mount:        /mnt/.__seed_device  
Blockmaps read:           1  
Seeding Dictionary updates: 957  
Streams read:             171615  
Comp bytes read:          4735312343  
Streams committed:        160674  
Streams deduped:          100181  
DS's committed:          368  
Total bytes processed:    7808325672  
Total bytes deduped:      2859079806  
Total inline bytes:       0  
Total orig bytes committed: 4634642077  
Total comp bytes committed: 4634642077  
Device orig bytes committed: 4634642077  
Device comp bytes committed: 4634642077  
Logical Avg Throughput:   134985.311 KB/s  
Logical Max Throughput:   339862.265 KB/s  
Physical Avg Throughput:  109999.312 KB/s  
Physical Max Throughput:  151030.517 KB/s  
Estimated time to sync:   0 days 0 hours 0 minutes 22 seconds  
  
Seed data progress:  
File Path:  Processed:  Pending:  Data to Device:  
  
seed_container:/file: 7808301653 Bytes: 2677458347 Bytes: 4949245866 Bytes  
  
Seed data history:  
File Path:  Processed:  Data to Device:  Throughput:
```

- 7 If seeding must be stopped for some time, or the device must be removed, you can execute the following command to stop all I/O to the seeding device.

```
seed -stop
```

```
[root@DR4000-DHYLBM1 ~]# seed --stop  
Successfully stopped seed job.
```

- 8 You can enter the following command to remove the device from the DRSeries system to which the seed data has been written. It is recommended that you execute the command, `seed -stop` before removing the device.

```
seed -remove_device
```

```
[root@DR4000-DHYLBM1 ~]# seed --remove_device  
Successfully deleted device details.
```

- 9 If the data to be seeded is completely gathered and the seed job is complete, you can delete the job by entering the following command.

```
seed -delete
```

```
[root@DR4000-DHYLBM1 ~]# seed --delete  
Successfully deleted seed job details.
```

## Important notes about the seeding export process

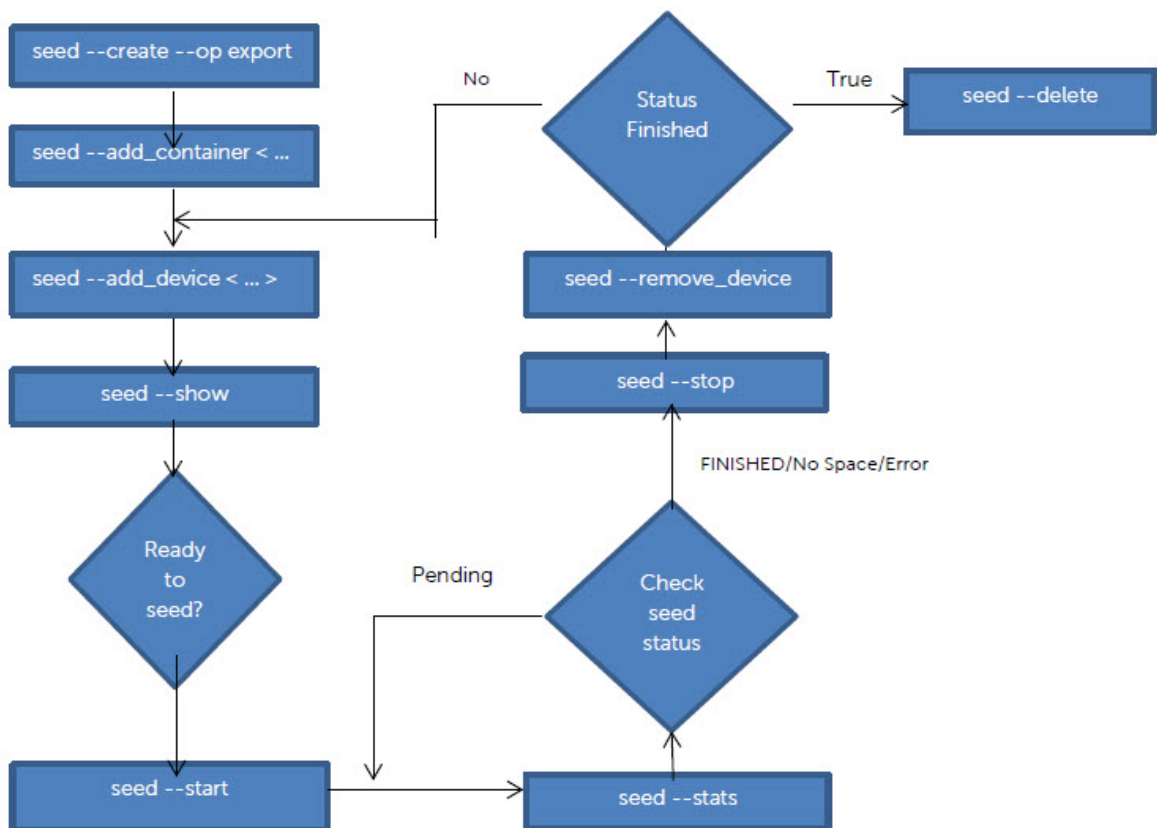
- If `'seed -delete'` is executed, all of the information about the seeding job is removed from the DR Series system. If a new job is created for the same container, then it will copy all the data once again.
- The seeding process does not require that data ingest be stopped; even to the container to which the data is seeded. This means the DRSeries system can continue to function normally while seeding is in process.
- If the external device runs out of space, you should enter the commands `'seed -stop,'` `'seed - remove_device,'` and then `'seed -add_device'` to add the new device. Enter `'seed -start'` to continue seeding to the new device or the same device after the data on the device is imported on the target.
- If the seeding process encounters any errors writing to the device, then you should remove the device by executing the command, `'seed -stop,'` followed by `'seed - remove_device'`. The device can be re-attached, or a new device can be attached. To start seeding after adding a device, enter the command, `'seed -start'`. Seeding at this point will continue from the point where it stopped earlier. If seeding is stopped for any

reason, enter the 'seed -stop' command, which will stop all of the seeding I/O activity. To continue again, enter the command, 'seed -start'.

- You need to export the attached device through the CIFS protocol. It can be a Windows share or a share exported from a Linux machine using the Samba protocol.
- During seeding, replication can be enabled as needed on the same containers to the same DRSeries system or to different DRSeries systems.
- Seeding does not delete the data on the device. Data on the device can also co-exist with the seed data. It is optional to delete the data on the device to free up the space for accommodating seed data.
- You are not required to seed the entire data. You can choose the amount of data that can be seeded, depending on factors like time, amount of external space available, and so on; and you can stop seeding accordingly. The residual data will reach the target during replication re-sync.

## Export CLI flow

The following diagram demonstrates the flow of the data export using the DRSeries system CLI commands.



## Setting up a seeding import job on the target DR Series system

The seed import process copies the de-duplicated data from the external device provided to the target DR Series system. The following steps describe the CLI commands you need to enter for the import job on the target DR Series system.

- 1 Create a seed import job using the following command.

```
seed --create --op import
```

This command requests a password, which is the same password defined at the time of data export. The password is used to verify and decrypt the data.

```
[root@swwsys-92 ~]# seed --create --op import
Enter password for seed import:
Re-enter password for seed import:
Successfully created seed job details.
```

- 2 You can reference the device from which the seed data is to be read by using the following command. (Note that only the shares exported by the CIFS protocol are recognized.)

```
seed --add_device --server <server name> --volume <volume> --username
<user name> [--domain <domain name>]
```

In this command, the parameters are as follows:

- **server name** - the name of the client that exports the device as a CIFS share.
- **volume name** - the name of the export share on the target server. The name is case sensitive.
- **user name** - the credentials needed to access the share. The password is requested separately.

```
[root@swwsys-92 ~]# seed --add_device --server 10.250.242.213 --volume Win_share --username administrator
Enter password for administrator@10.250.242.213:
Successfully added seed device.
```

- 3 Once the device is added successfully, you can cross check the configuration details by using the following command. This step is recommended before you proceed further.

seed --show

```
[root@swwsys-92 ~]# seed --show
Device info
=====
Server                :10.250.242.213
Volume                :Win_share
Username              :administrator
Domain                :NA

Job info
=====
Operation              :Import
Status                 :Initialized
Encryption type        :aes256
```

- 4 To start the seeding process, enter the following command.

seed --start

The data copy begins and continues until all of the data from the device is completely read.

```
[root@swwsys-92 ~]# seed --start
Successfully started seed job.
```

- 5 To monitor the seeding process, enter the following command.

seed --stats

The type of information in the output of the command, “seed --stats,” looks similar to the following example.

```
[root@swwsys-92 ~]# seed --stats
Seeding Target Stats:

Seed state:          SEED_STARTED
Seed status:         PENDING
Seed device mount:   /mnt/.__seed_device
Seed export host:    DR4000-DHYLBM1.ocarina.local
Seed export device added at: 2014-08-18 00:21:16
DS's read:          40
Streams read:        17435
ZL entries written: 0
Dictionary updates: 0
Dup streams:         17435
Bytes deduped:       504068720
Streams committed:  0
DS's committed:     0
Total orig bytes committed: 0
Total comp bytes committed: 0
Avg Throughput:      56378.668 KB/s
Max Throughput:      100843.872 KB/s
Estimated time to sync: 0 days 0 hours 4 minutes 21 seconds
```

If the seed status is 'FINISHED,' then the data has been completely read from the attached device attached. The device can be removed after issuing the command, 'seed -stop,' and a new device if present can be attached.

- 6 If you need to stop seeding, or the device needs to be removed, you can enter the following command to stop all I/O from the seeding device.

```
seed -stop
```

```
[root@swwsys-92 ~]# seed --stop
Successfully stopped seed job.
```

- 7 To remove the device from the DR Series system to which the seed data has been written, enter the following command. (It is recommended that you enter the command, `seed -stop`, before removing the device.)

```
seed --remove_device
```

```
[root@swwsys-92 ~]# seed --remove_device
Successfully deleted device details.
```

- 8 If the data gathering from all the devices is complete, you can enter the following command to remove the seed job completely.

```
seed --delete
```

```
[root@swwsys-92 ~]# seed --delete
Successfully deleted seed job details.
```

- 9 Once the import is complete, replication re-sync must be performed between the source and target DR Series system containers. Once replication reaches the 'INSYNC' state, execute the following command to remove any data that is left unaccounted from seeding.

```
seed --cleanup
```

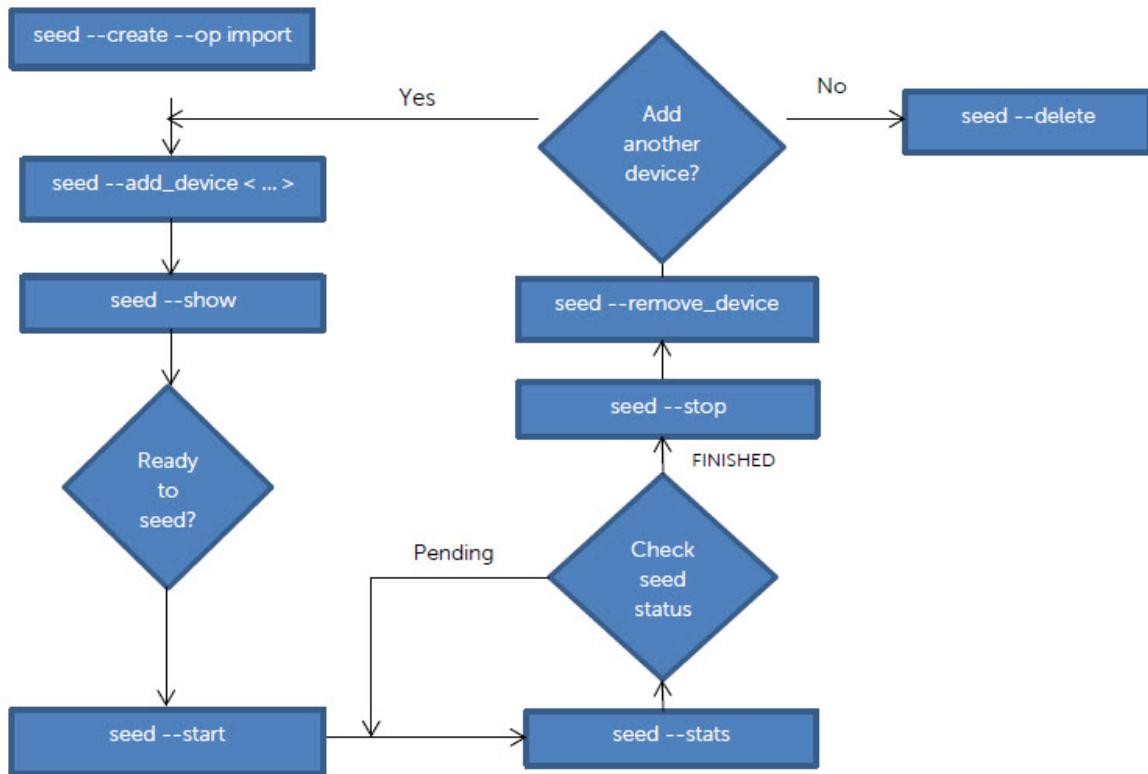
```
[root@swwsys-92 ~]# seed --cleanup  
Successfully added seed ZL logs to cleaner queue.
```

## Important notes about the import seeding process

- The order in which the seed devices are attached makes no difference for the seeding process.
- The data at source could have changed after the export process is initiated. This data is replicated during replication re-sync.
- The command 'seed --cleanup' must be executed only after the replication is established between source and target. If it is executed before, the entire imported data set could be deleted.
- If seeding has to be stopped for any reason, you should enter the 'seed --stop,' which will stop all seeding I/O activity. To continue, enter the command, 'seed --start'. If an error occurs while reading the seeding device, you should remove the device using the command, 'seed --remove\_device,' after stopping the I/O, and then re-attach the device.
- If any device is lost, corrupted, or left unattached during the import process, the data in that particular device will be received at the time of replication resync and will not have any effect on the import process.
- Once the data on the device is imported, the seed import process will not delete the data on the seed location. You could use the same data, or the device, to import the data on a different DR Series system if needed (for example, to setup cascaded replication). However, it is recommended that you remove the data after replication setup is complete.

# Import CLI flow

The following diagram demonstrates the flow of the data import using the DR Series system CLI commands.





---

## Synchronizing the source and target containers

Once seeding import is complete, two possible scenarios exist:

- The target container already exists, or,
- The target container does not exist, and, therefore, a fresh replication pair must be created.

### If a target container exists

If a target container exists, you should initiate replication re-sync between the source and target DR Series system containers. Since the data is already present on the target, the re-sync will complete quickly after transferring the namespace with any other new or changed data. Complete the following steps.

- 1 Enter the command:  

```
replication --resync --name <name> --role <source | target>
```
- 2 Wait for replication state to come to "INSYNC".

### If the target container does not exist

If a target container does not exist, complete the following steps.

- 1 Create a container on the target DR Series system.
- 2 Enable replication between the source and target DR containers by using the following command:  

```
replication --add < ... >
```

```
Replication Source Container : seed_container
Replication Source System   : 10.250.242.157 (DR4000-DHYLBM1)
Replication Target Container : seed_container
Replication Target System   : 10.250.240.157 (dr4000-4zkrbm1)
Peer Status                 : Online
Replication State           : INSYNC
Schedule Status             : In window
Replication Average Throughput : 0 KiB/s
Replication Maximum Throughput : 73401 KiB/s
Network Average Throughput   : 0 KiB/s
Network Maximum Throughput   : 42637 KiB/s
Network Bytes Sent           : 7.29 GiB
Pending Bytes                : 0.00 B
Dedupe Network Savings      : 25.72%
Compression Network Savings  : 0.00%
Last INSYNC Time            : 2014-08-12 09:16:50
Estimated Time to Sync      : 0 days 0 hours 0 minutes 0 seconds
```

- 3 Once the replication status changes to 'INSYNC,' then both source and target have the same data.
- 4 Enter the following command on the target to remove unwanted data created by seeding:  
`seed -cleanup`