

**7MTT – Oracle 7mode to CDOT Work Instructions**

**Authors:** David Ng

**Contributors:**

**Document Version:** V2

**Date:** April 2015

Status: Published

**CONFIDENTIAL INFORMATION**

This document contains information proprietary to Thomson Reuters and may not be reproduced, disclosed or used in whole or part without express permission of Thomson Reuters.

© Thomson Reuters 2013

Contents

[1 Introduction 3](#_Toc430699729)

[1.1 Management Summary 3](#_Toc430699730)

[1.2 Change History 3](#_Toc430699731)

[1.3 Distribution List 3](#_Toc430699732)

[1.4 Glossary 3](#_Toc430699733)

[2 7Mode to CDOT Migration 5](#_Toc430699734)

[2.1 High Level Steps (DB Refresh) 5](#_Toc430699735)

[2.2 High Level Steps (Migration) 5](#_Toc430699736)

[2.3 Pre-requisites 6](#_Toc430699737)

[2.4 Vserver and LIF creation 6](#_Toc430699738)

[2.4.1 Create Vserver (replace hyphen with underscore in vserver rootvolume name) 6](#_Toc430699739)

[2.4.2 Create LIF with default route and failover group 6](#_Toc430699740)

[2.5 DNS configuration 7](#_Toc430699741)

[2.5.1 Setup DNS on a Vserver 7](#_Toc430699742)

[2.6 Showmount script user 7](#_Toc430699743)

[2.7 Oracle account setup 7](#_Toc430699744)

[2.7.1 Oracle role, user, and SSH publickey configuration (ssh keys are located in DFM server ‘/filers/admin/source/logical’) 7](#_Toc430699745)

[2.8 NFS and CIFS configuration 8](#_Toc430699746)

[2.8.1 Enable NFSv3 8](#_Toc430699747)

[2.9 Start the Migration 8](#_Toc430699748)

[2.10 Primary job schedule and snapshot policy configuration 26](#_Toc430699749)

[2.10.1 Create job schedule and snapshot policy (only execute for OraAdmin volume): 26](#_Toc430699750)

[2.11 Volume creation 26](#_Toc430699751)

[2.11.1 Create export policies (repeat this step for each volume) 26](#_Toc430699752)

[2.11.2 Create the default export-policy (repeat this command for every nfs client) 26](#_Toc430699753)

[2.11.3 Create export policy rule (repeat this command for each volume and nfs client) 26](#_Toc430699754)

[2.11.4 Add volumes to export & snapshot policies (execute for oraadmin): 26](#_Toc430699755)

[2.11.5 Add volumes to export & snapshot policies (execute for \*ora\*\_nosnap & \*ora\*\_snap): 26](#_Toc430699756)

[2.11.6 Delete snapshot policies which were created by 7MTT (names start with ‘SnapShot\_\*): 27](#_Toc430699757)

[2.11.7 Rename junction paths (execute for each volume) 27](#_Toc430699758)

[2.11.8 Setup snap autodelete on volumes (run for each SNAP volume) 27](#_Toc430699759)

[2.11.9 Create qtrees quotas (repeat command below for each required qtree) 27](#_Toc430699760)

[2.11.10 Enable the quotas (repeat step for each volume> 27](#_Toc430699761)

[2.12 QoS policy group creation 27](#_Toc430699762)

[2.12.1 Create QoS policy group and apply it at the volume level (create one QOS policy for each volume) 27](#_Toc430699763)

[3 Snapvault Configuration 28](#_Toc430699764)

[3.1 Cluster and Vserver Peering 28](#_Toc430699765)

[3.1.1 Confirm that cluster peering has been enabled 28](#_Toc430699766)

[3.1.2 Create the cluster peer (skip this step if cluster peering has been configured) 28](#_Toc430699767)

[3.1.3 Confirm if vserver peering has been configured 28](#_Toc430699768)

[3.1.4 Create vserver peering on the destination system (skip this step if vserver peering has been configured) 28](#_Toc430699769)

[3.1.5 Accept the vserver peering on the source system 28](#_Toc430699770)

[3.2 SnapVault configuration 28](#_Toc430699771)

[There are two data protection configurations for LION: SNAP & NOSNAP. The ‘oradata’ and ‘oraadmin’ volume are vaulted in a SNAP configuration. Only the ‘oraadmin’ volume is vaulted in a NOSNAP configuration. Hence, the number of volumes you execute commands for may vary based on configuration type. 28](#_Toc430699772)

[3.2.1 Create secondary volumes for SnapVault as type “DP” on the destination cluster 28](#_Toc430699773)

[3.2.2 Create a cron job schedule if it does not exist in the destination 29](#_Toc430699774)

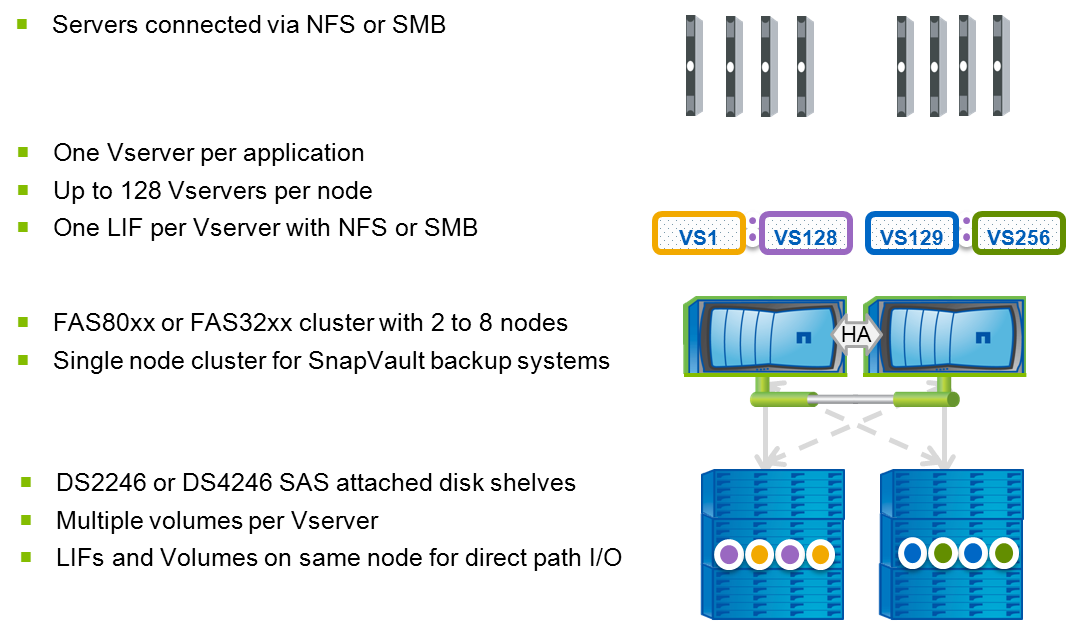
[3.2.3 Configure a snapmirror policy on the destination (execute these command for oraadmin & oradata) 29](#_Toc430699775)

[3.2.1 Initialize SnapVault relationship on the destination (execute these command for oraadmin & oradata) 29](#_Toc430699776)

# Introduction

## Management Summary

This document details the process used to migrate an Oracle Database from 7mode to CDOTwith 7MTT.



## Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver** | **Date** | **Author** | **Key Changes** |
| 1 | April 2015 | David Ng | Initial Version |

## Distribution List

|  |  |
| --- | --- |
| **Name** | **Role** |
| Storage Support |  |
| Storage Delivery |  |

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| cDOT | clustered Data ONTAP |
| Vserver | A logical storage virtual server, also known as a Storage Virtual Machine (SVM), which contains LIFs, Volumes, and configuration information such as access control details. |
| LIF | Logical Interface – a cDOT logical network interface with an IP address, assigned to a single Vserver. |
| 7MTT | 7mode Transition Tool – A tool used to migrate from 7mode to CDOT |
| WFA | OnCommand Workflow Automater – An automation framework application from NetApp, used for storage provisioning. |

# 7Mode to CDOT Migration

## High Level Steps (DB Refresh)

|  |  |
| --- | --- |
| **DB Tech Refresh Steps (7mode to CDOT)** | **Step Owner** |
| Install new database in target system | DBA |
| shut down; save off system, ctl files, etc. as needed on target | DBA |
| unmount s\*ora\*\_snap from target | Unix |
| Create new SVM & volume | Storage |
| initialize replication to existing s\*ora\*snap volume, or create new s\*ora\*\_snap volume and initialize replication; rename old volume with expiration date | Storage |
| Disable monitoring during maintenance window | Application/DBA |
| Stop Application/Database during maintenance window | Application/DBA |
| Unmount source volumes during maintenance window | Unix |
| Final snapmirror sync/break during maintenance window | Storage |
| Update '/etc/fstab' during maintenance window | Unix |
| Mount new volumes during maintenance window | Unix |
| Restore Database, move system, ctl, etc files into place during maintenance window | DBA |
| Follow standard TTS import of tablespaces that were mirrored over during maintenance window | DBA |
| Start DB during maintenance window | DBA |
| Reboot Server & verify environment during maintenance window | Unix/DBA |

## High Level Steps (Migration)

|  |  |
| --- | --- |
| **DB Migration Steps (7mode to CDOT)** | **Step Owner** |
| Create SVM & volumes | Storage |
| Initialize volume replication before maintenance window | Storage |
| Disable monitoring during maintenance window | Application/DBA |
| Stop Application/Database during maintenance window | Application/DBA |
| Unmount source volumes during maintenance window | Unix |
| Final snapmirror sync/break during maintenance window | Storage |
| Update '/etc/fstab' during maintenance window | Unix |
| Mount new volumes during maintenance window | Unix |
| Start DB and verify no file read/write errors during maintenance window | DBA |
| Reboot Server & verify environment during maintenance window | Unix/DBA |

## Pre-requisites

* Confirm the target aggregate has sufficient capacity
* Confirm the source volume is 64 bit
* Register the target vserver hostname in DNS
* Make note of service accounts on the source so they can be created in destination
* Confirm there is network connectivity between the source and target
* Add the CDOT vserver name (not IP) to the source systems /etc/snapmirror.allow file
* Make note of qtree quotas sizes so they can be applied in destination
* Make note of the source volume language
* Make note of the backup volume size

## ****Vserver and LIF creation****

### Create Vserver (replace hyphen with underscore in vserver rootvolume name)

vserver create -vserver <vsname> -rootvolume <vsname>\_root -aggregate <aggrname> -ns-switch file -nm-switch file -rootvolume-security-style unix -language <language>

vserver show

### Create LIF with default route and failover group

network interface create -vserver <vsname> -lif <vsname>-lif-<lif#> -role data -data-protocol nfs -home-node <node> -home-port <port> -address <ip> -netmask <netmask> -status-admin up -firewall-policy mgmt -failover-group <group>

network routing-groups route create -vserver <vsname> -routing-group d<network>/<mask> -destination 0.0.0.0/0 -gateway <gateway>

vserver show

network interface show

network interface show -failover

network routing-groups route show –vserver <vsname>

## ****DNS configuration****

### Setup DNS on a Vserver

vserver services dns create -vserver <vsname> -domains <domainname> -name-servers <comma\_separate\_name\_server\_list>

vserver services dns show

## Showmount script user

security login role create -role showmount -cmddirname "vserver export-policy" -access readonly -vserver <vserver>

security login role create -role showmount -cmddirname volume -access readonly -vserver <vserver>

security login role create -role showmount -cmddirname "version" -access all -vserver <vsname>

security login create -username shwmnt -application ontapi -authmethod password -role showmount -vserver <vserver>

## ****Oracle account setup****

### Oracle role, user, and SSH publickey configuration (ssh keys are located in DFM server ‘/filers/admin/source/logical’)

security login role create -role oracle -cmddirname "volume snapshot" -access all -vserver <vsname>

security login role create -role oracle -cmddirname "set" -access all -vserver <vsname>

security login role create -role oracle -cmddirname "version" -access all -vserver <vsname>

security login role create -role oracle -cmddirname "job show" -access readonly -vserver <vsname>

security login role create -role oracle -cmddirname "df" -access readonly -vserver <vsname>

security login role create -role oracle -cmddirname "snapmirror list-destinations" -access readonly –vserver <vsname>

security login create -username oracle -application ssh -authmethod publickey -role oracle -vserver <vsname>

security login publickey create -username oracle -vserver <vsname> -publickey "<ssh-dss pub\_key\_string>"

## ****NFS and CIFS configuration****

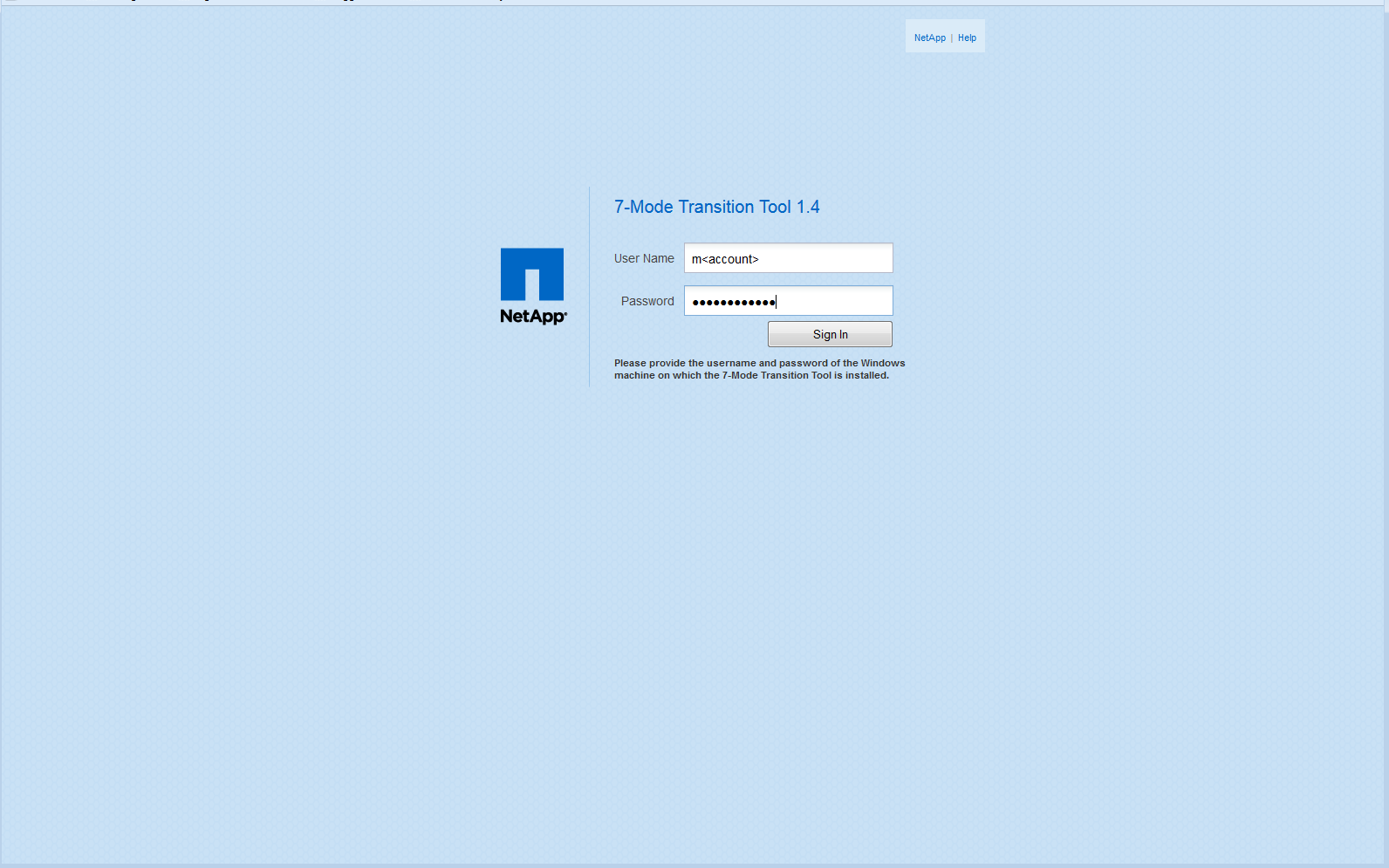
### Enable NFSv3

vserver nfs create -vserver <vsname> -access true -v3 enabled

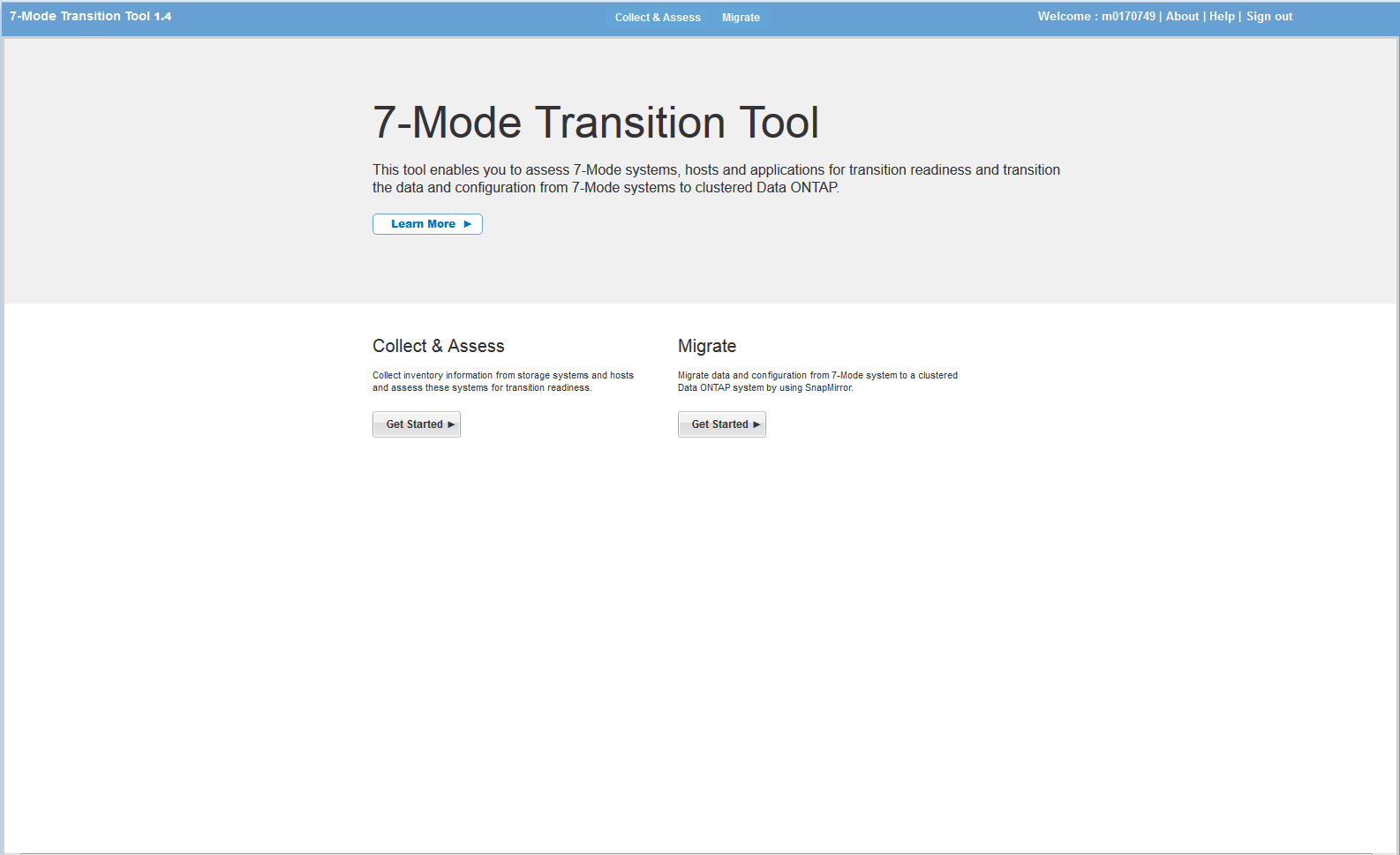
vserver nfs show

## ****Start the Migration****

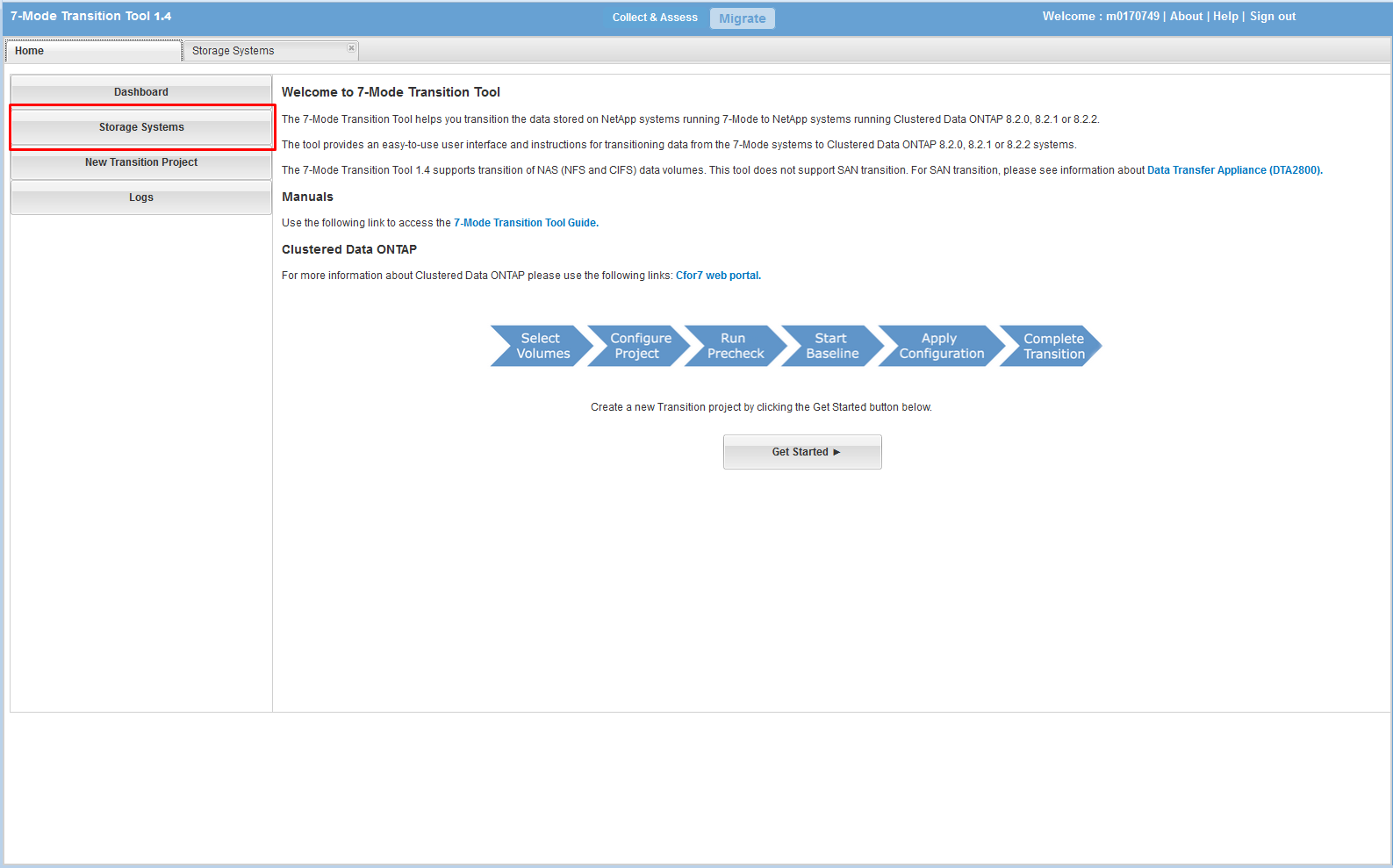
Connect the 7MTT (https://<7MTT\_Server>:8443/transition) tool via web browser



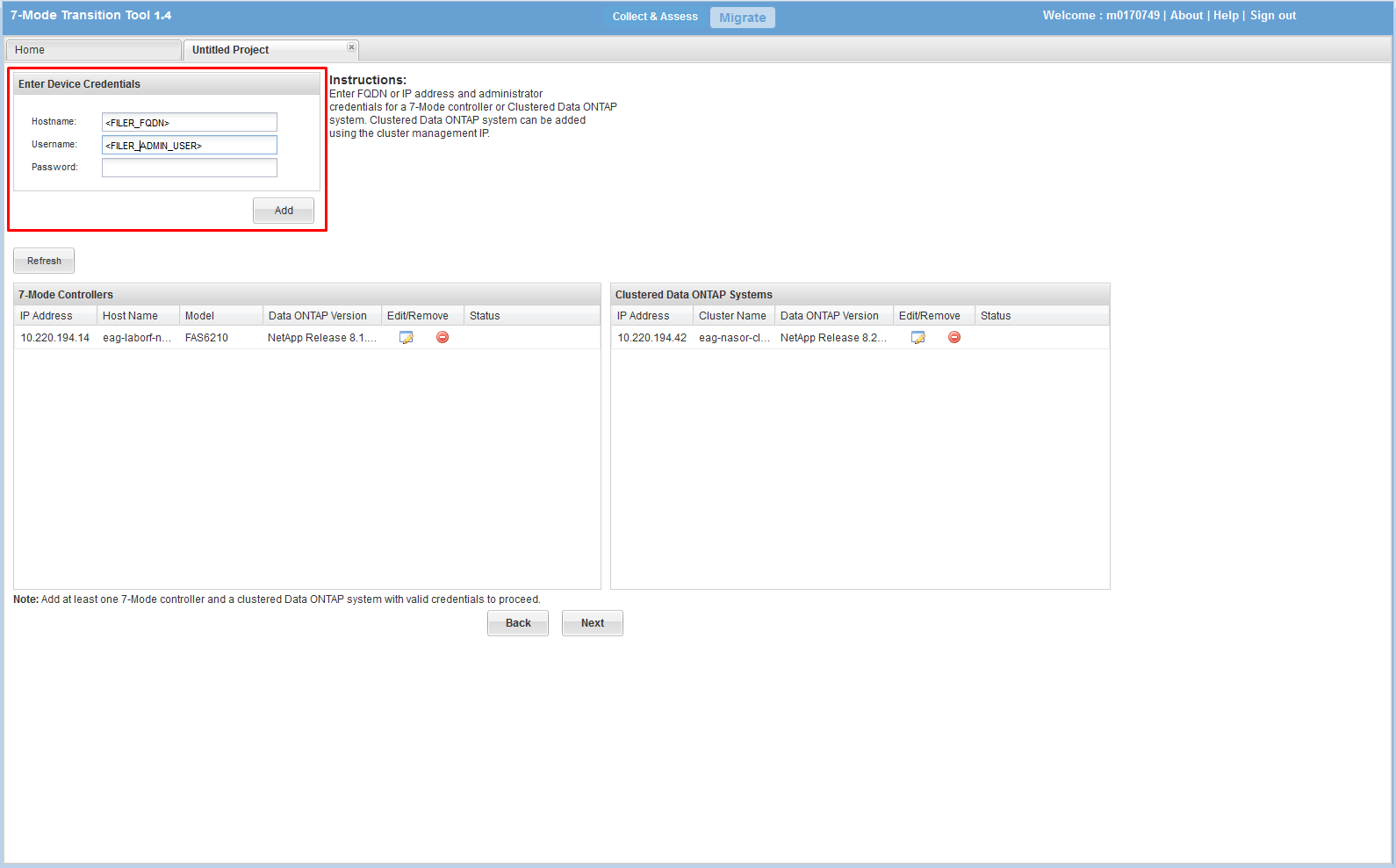
* Click ‘Migrate’



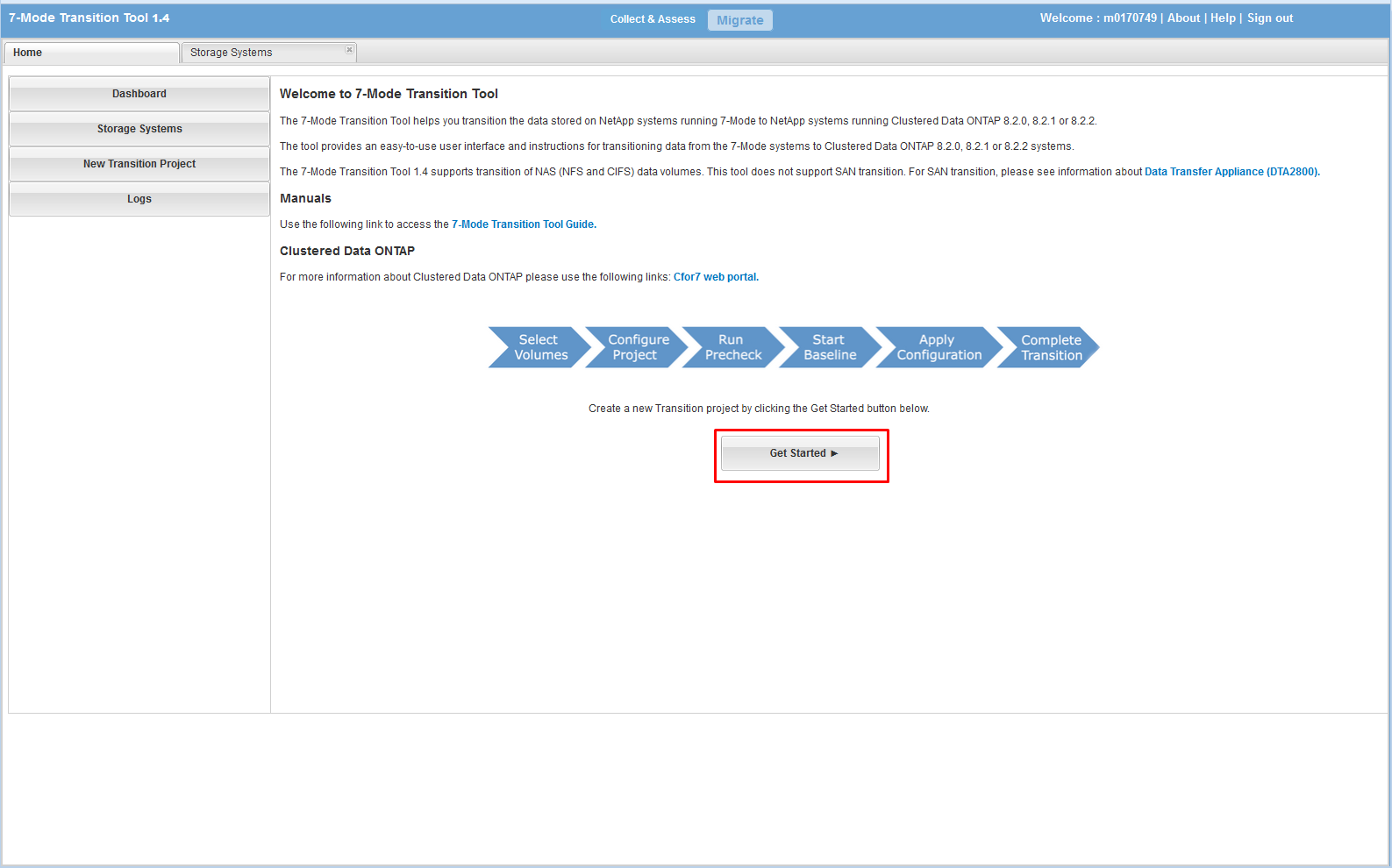
CLICK on the ‘Storage System’ button in the left pane:



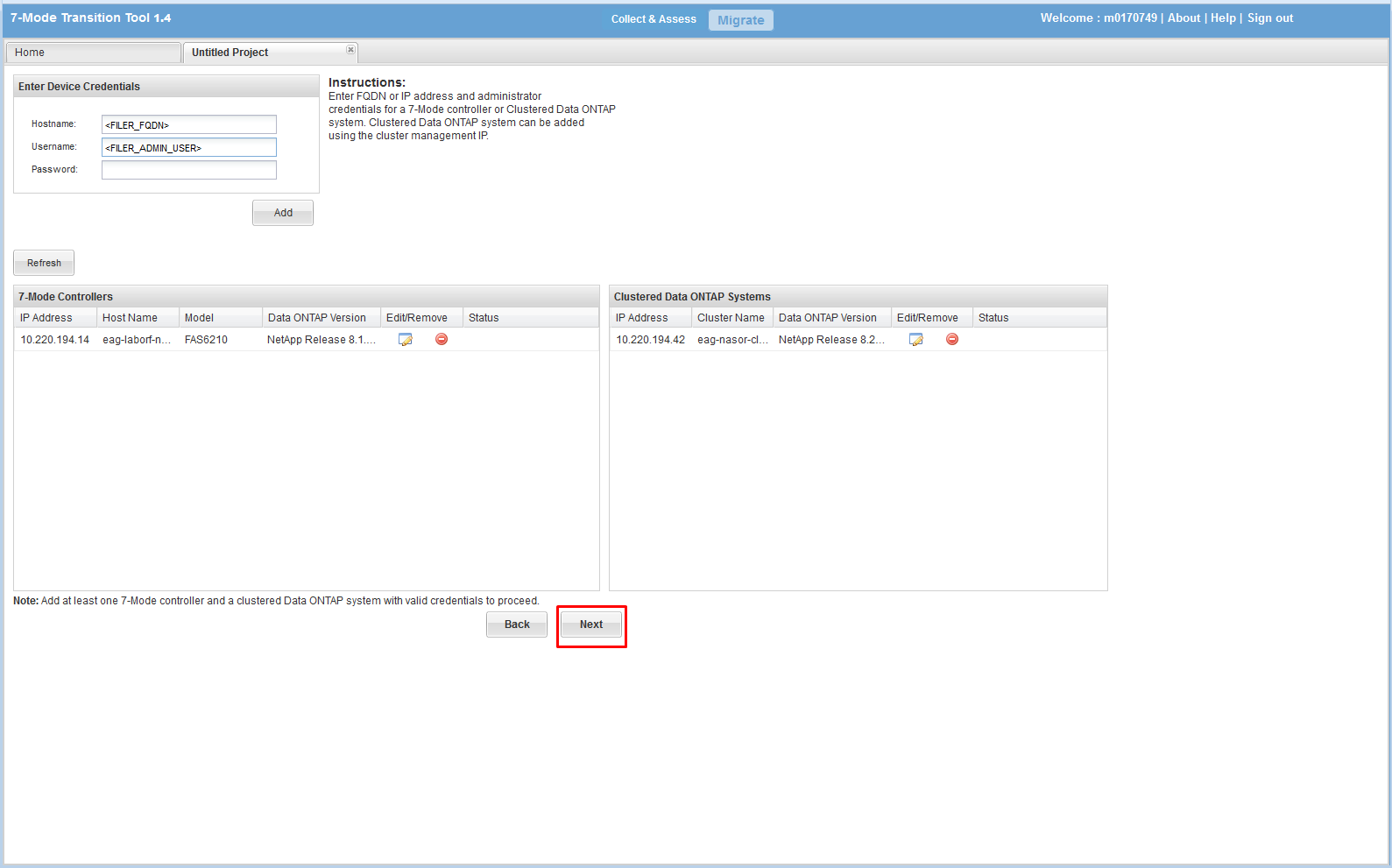
* Add storage systems to 7MTT (execute this step for the source and target storage systems):
* ENTER the FQDN of the filer
* CLICK ADD



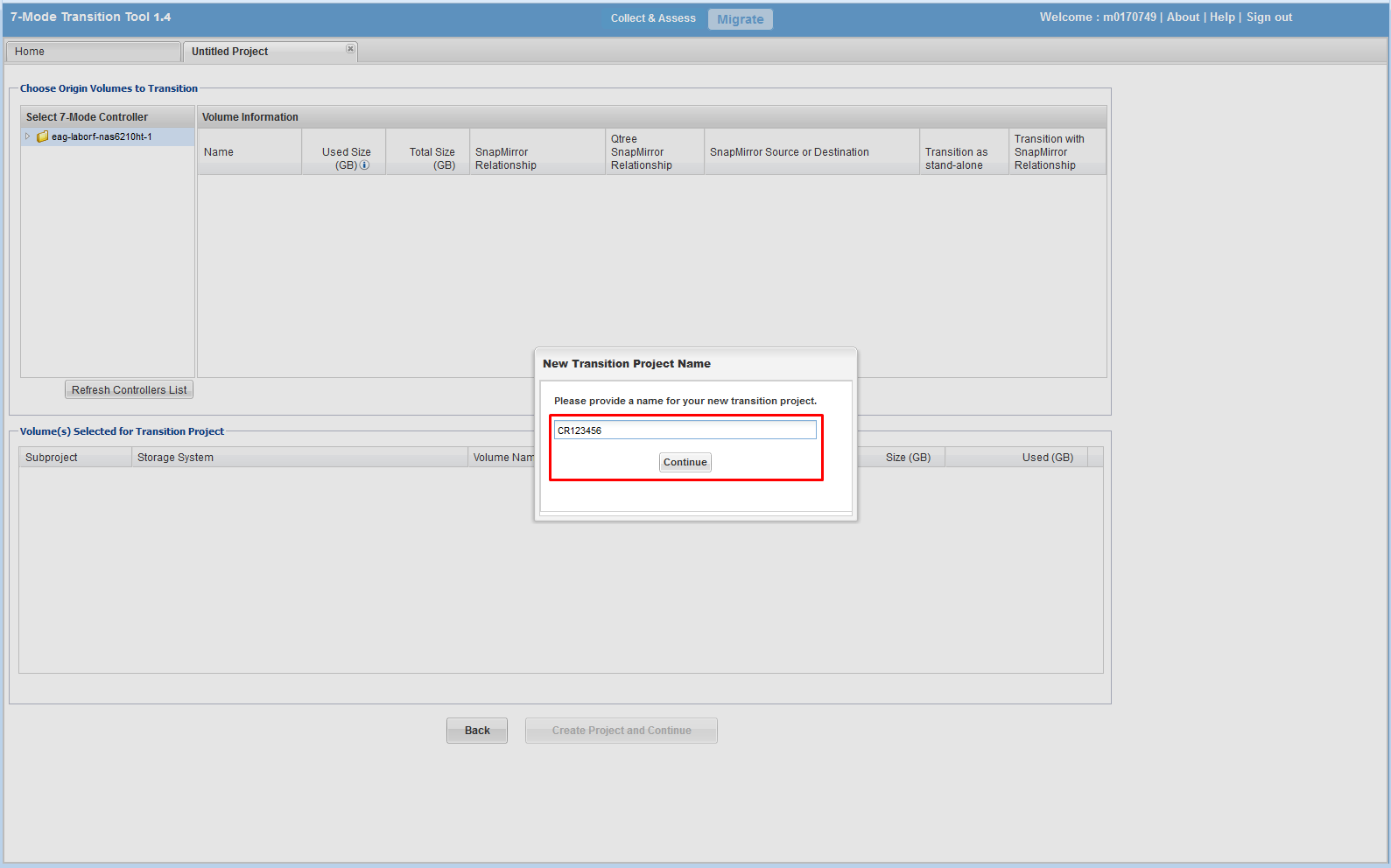
* CLICK the ‘HOME’ tab -> CLICK ‘Ge Started’



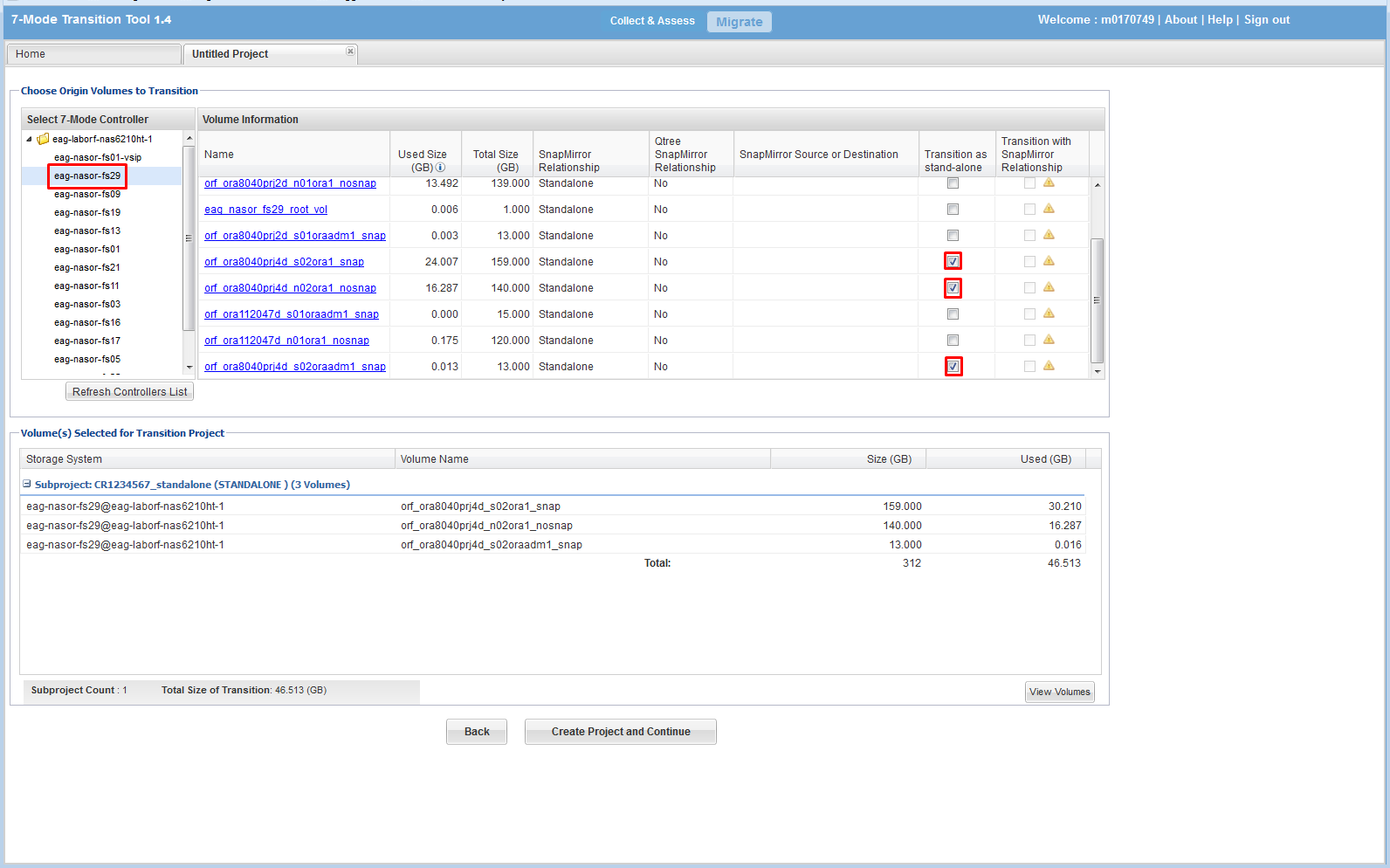
* Confirm that the source and destination storage systems are listed
* CLICK ‘Next’



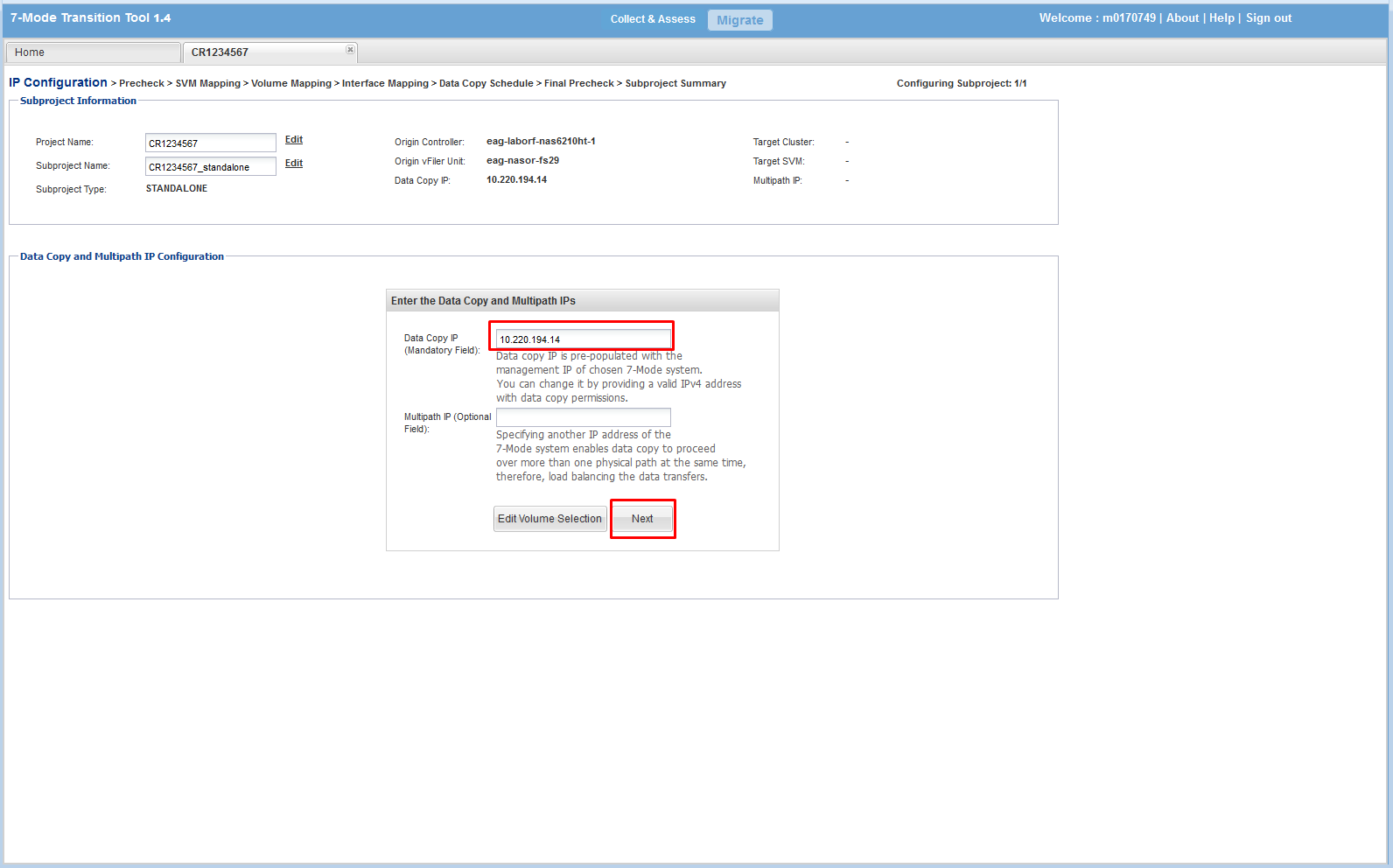
* ENTER a name for the project
* CLICK ‘Continue’



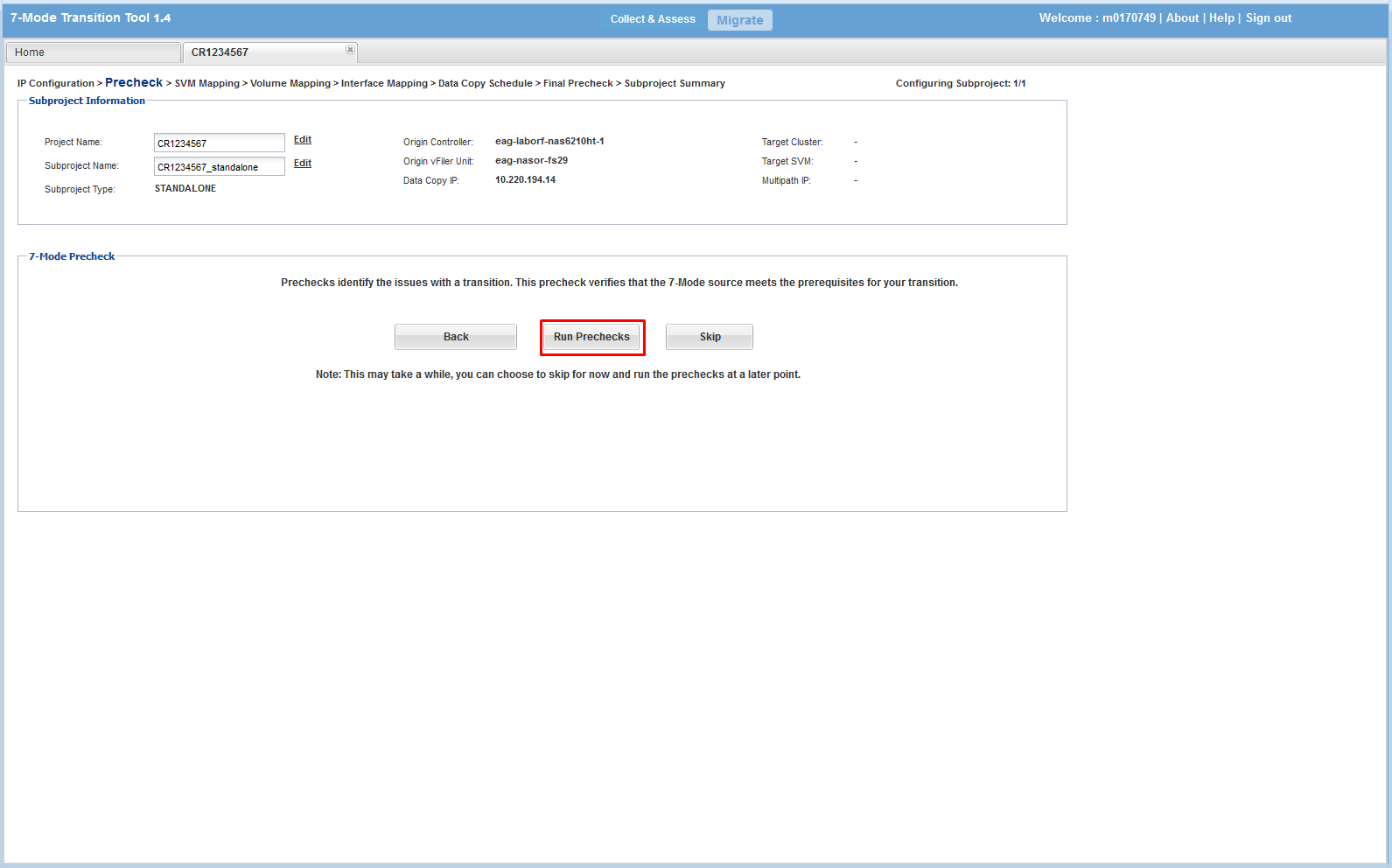
* SELECT the source vfiler and volume(s):
* CLICK ‘Create Project and Continue’



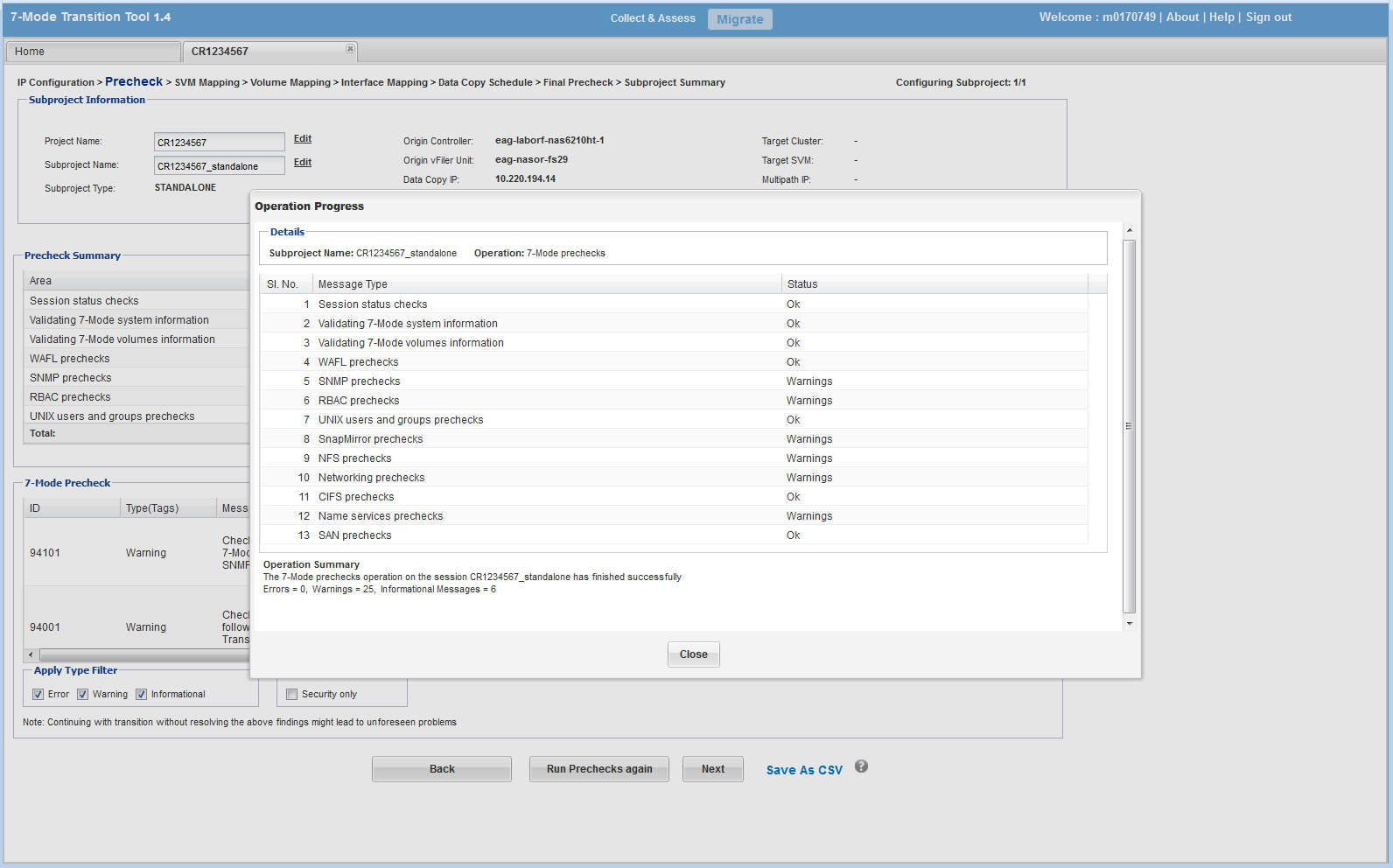
* ENTER an IP to be used for replication traffic on the source filer
* CLICK ‘Next’



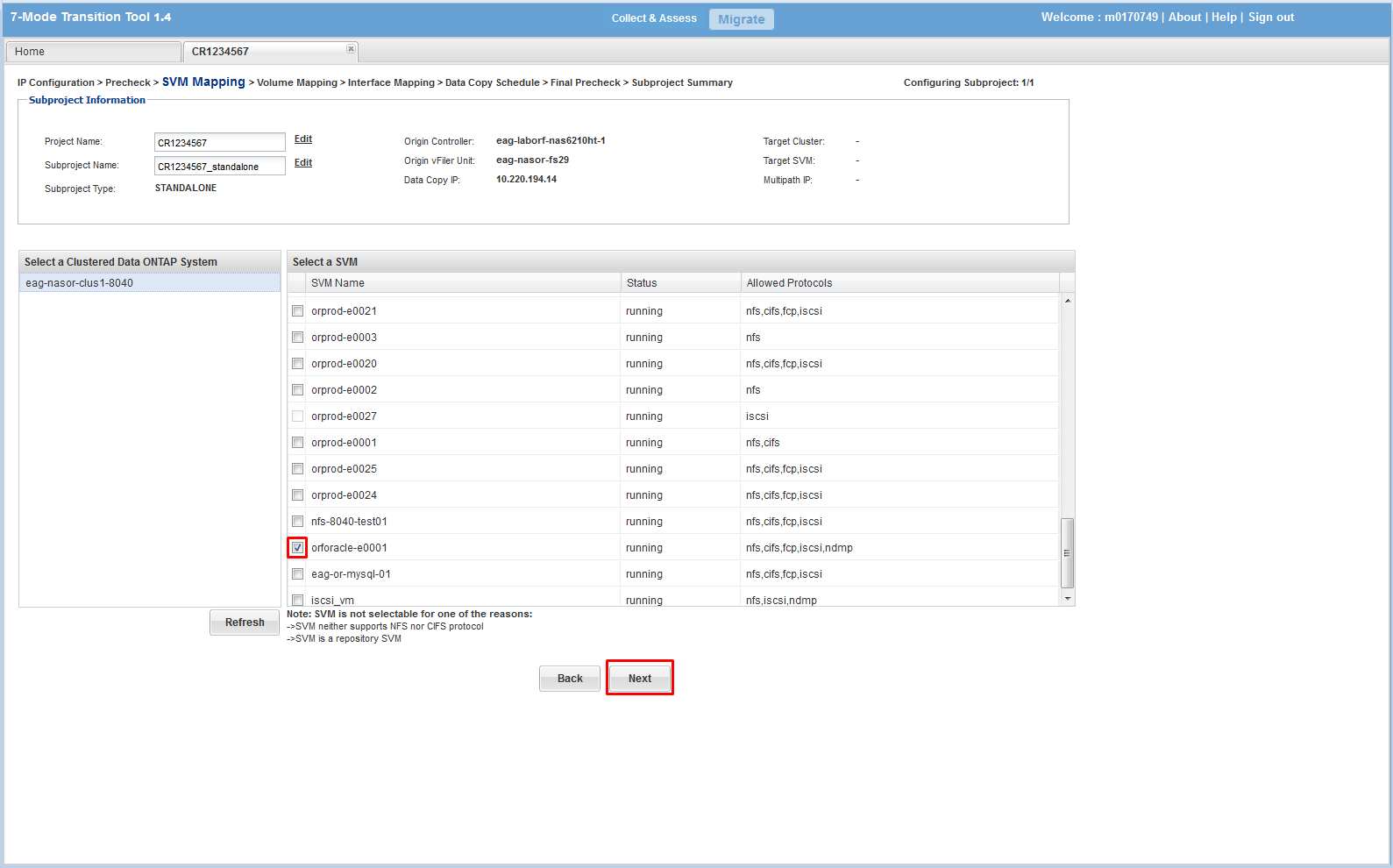
* The next window will ask if you want to run pre-check
* CLICK ‘Run Prechecks’



* Review the Report. You can ignore the warnings listed below. Resolve errors if they appear.
* CLICK ‘Close’
* CLICK ‘Next’

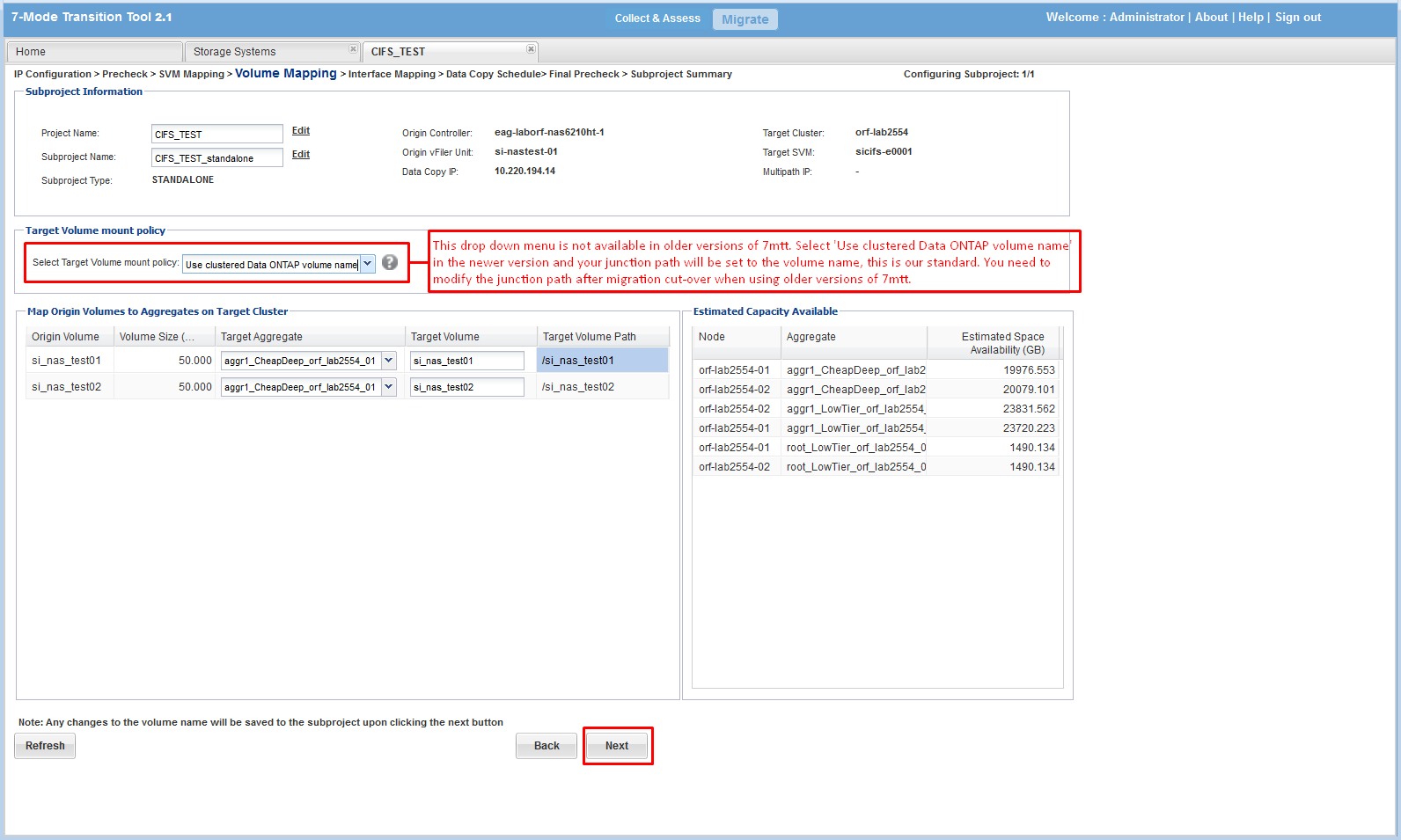


* SELECT a target vserver
* CLICK ‘Next’



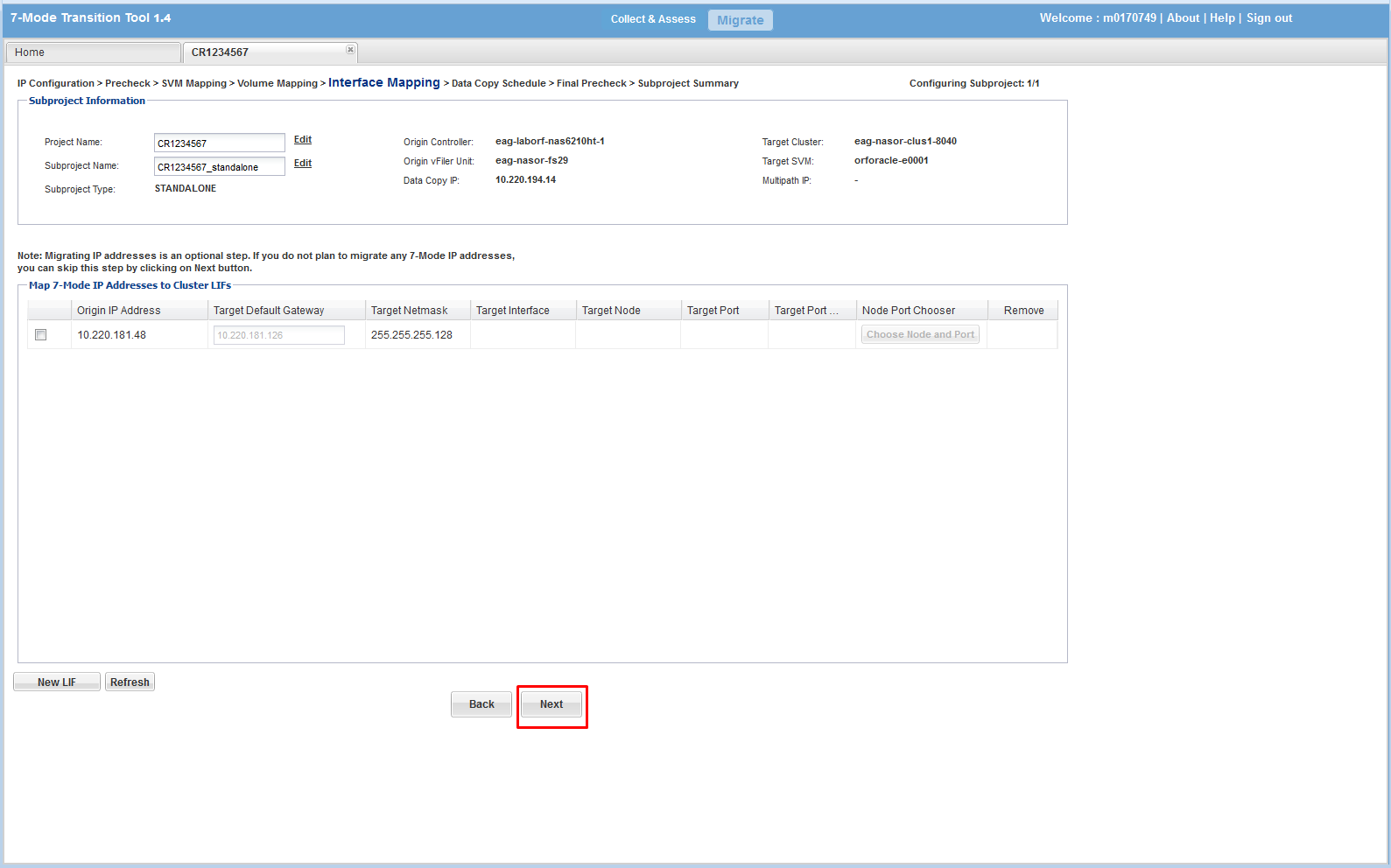
**IMPORTANT: Make sure you enter the correct information in this section! Newer versions of 7MTT allow you to set junction path = volume name, this is the DCO standard. But, you need to modify the junction-path after the cut-over manually in older versions of 7MTT.**

* ENTER the destination aggregate
* ENTER the destination volume name
* SELECT ‘Use Clustered OTAP volume name’ from ‘Target Volume Mount Policy’ if drop down menu is available
* CLICK ‘Next’

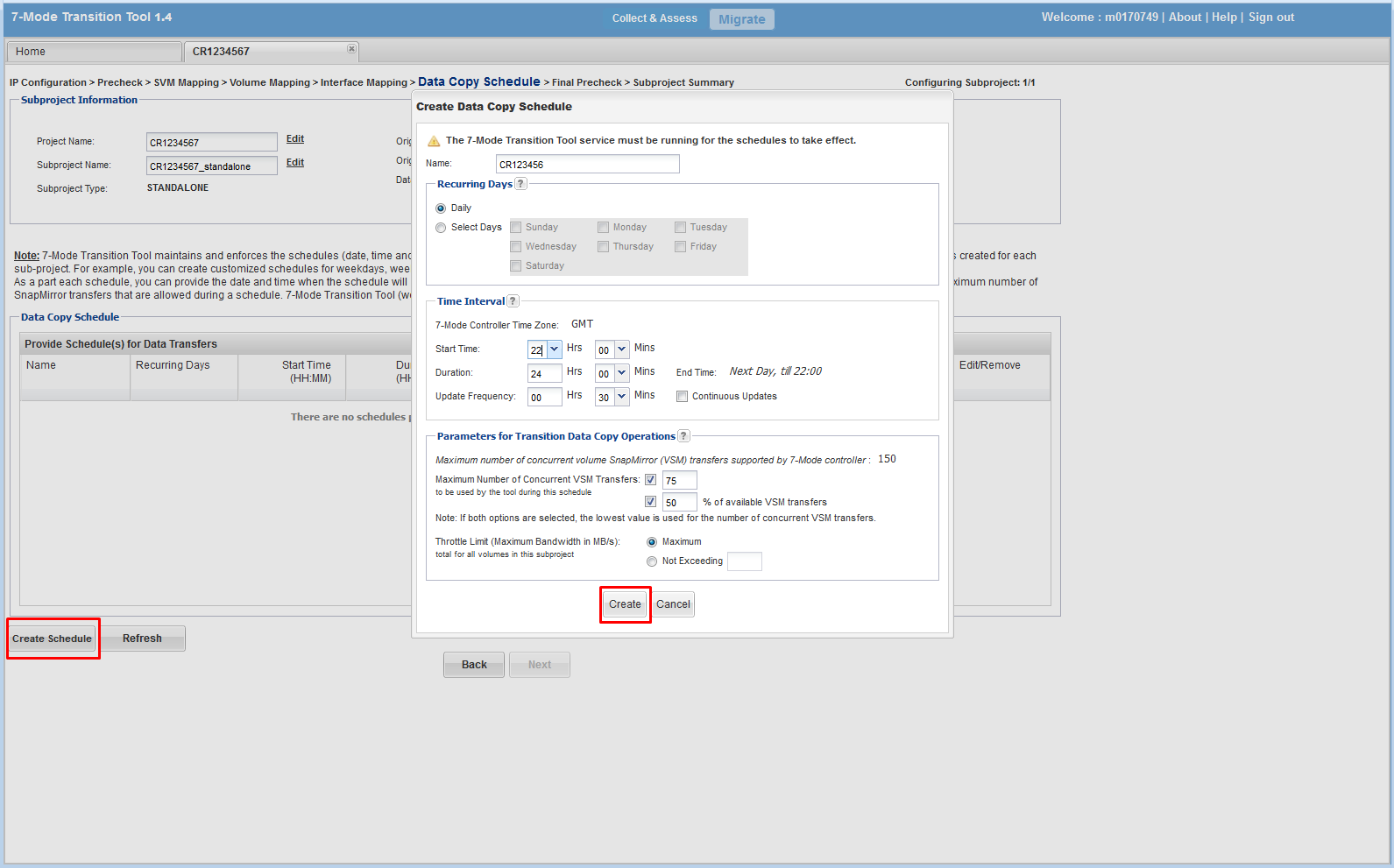


NOTE: We are not migrating IP addresses

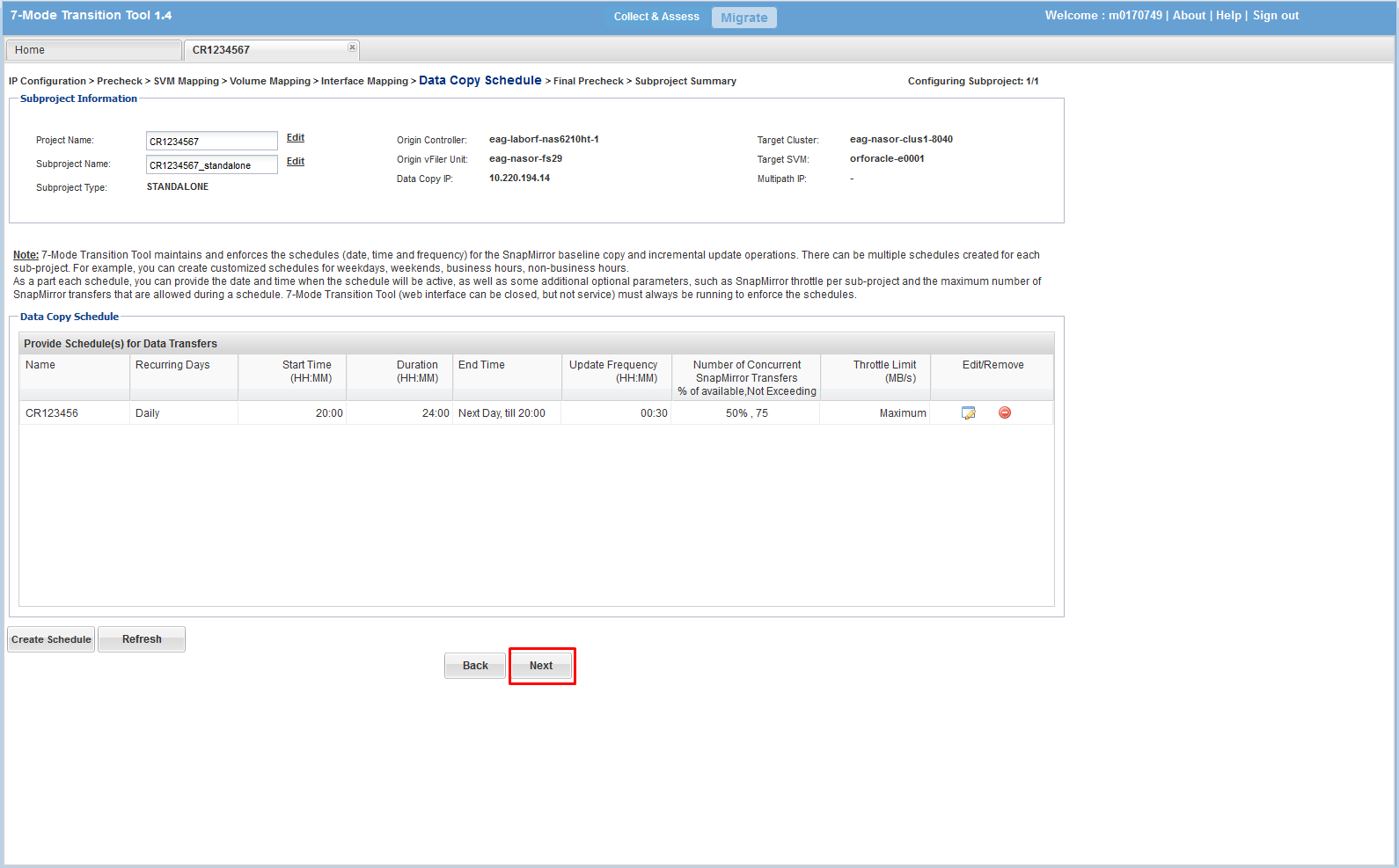
* CLICK ‘Next’



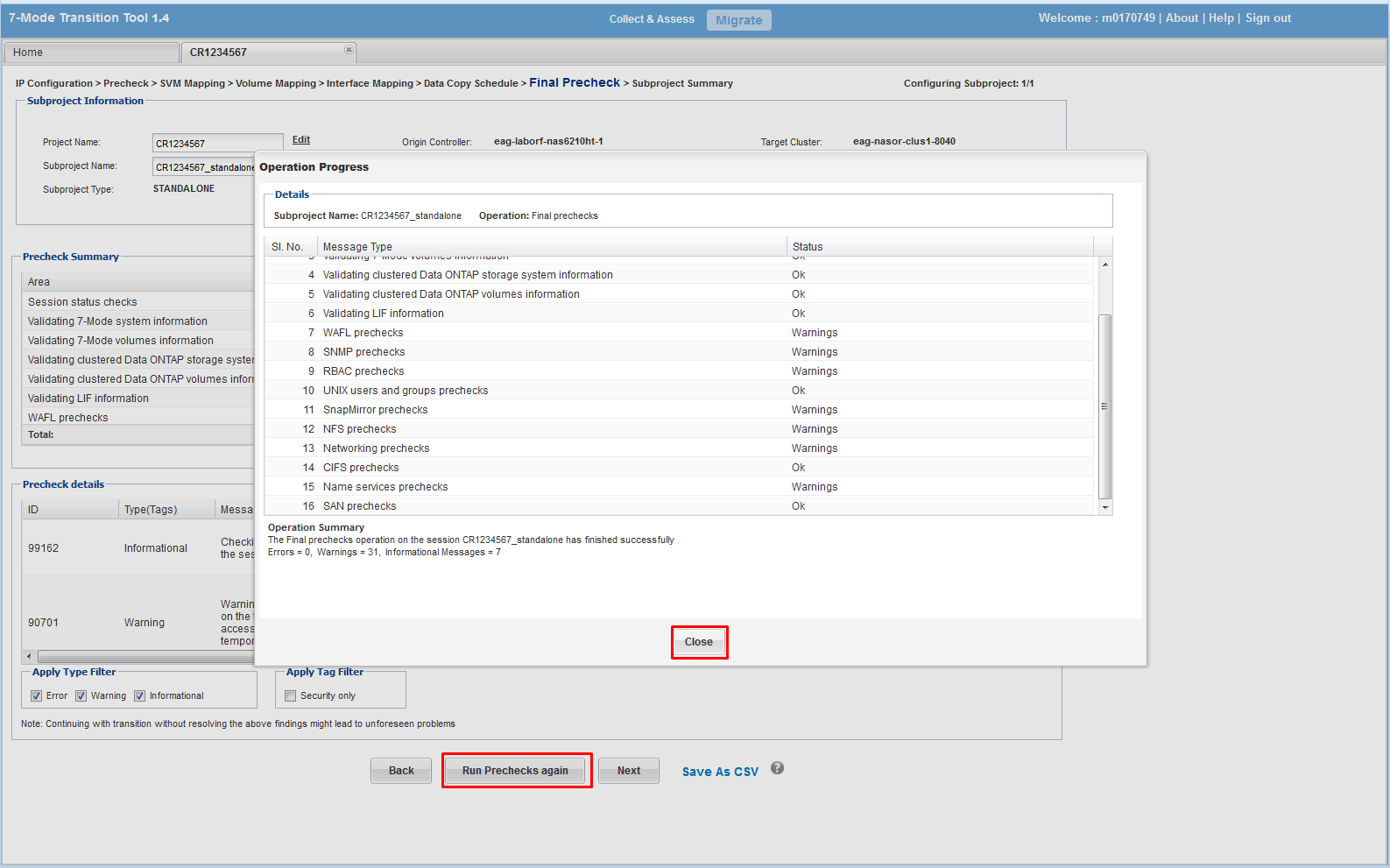
* CLICK ‘Create Schedule’, a popup box will appear
* ENTER a replication schedule (run off hours)
* CLICK CREATE



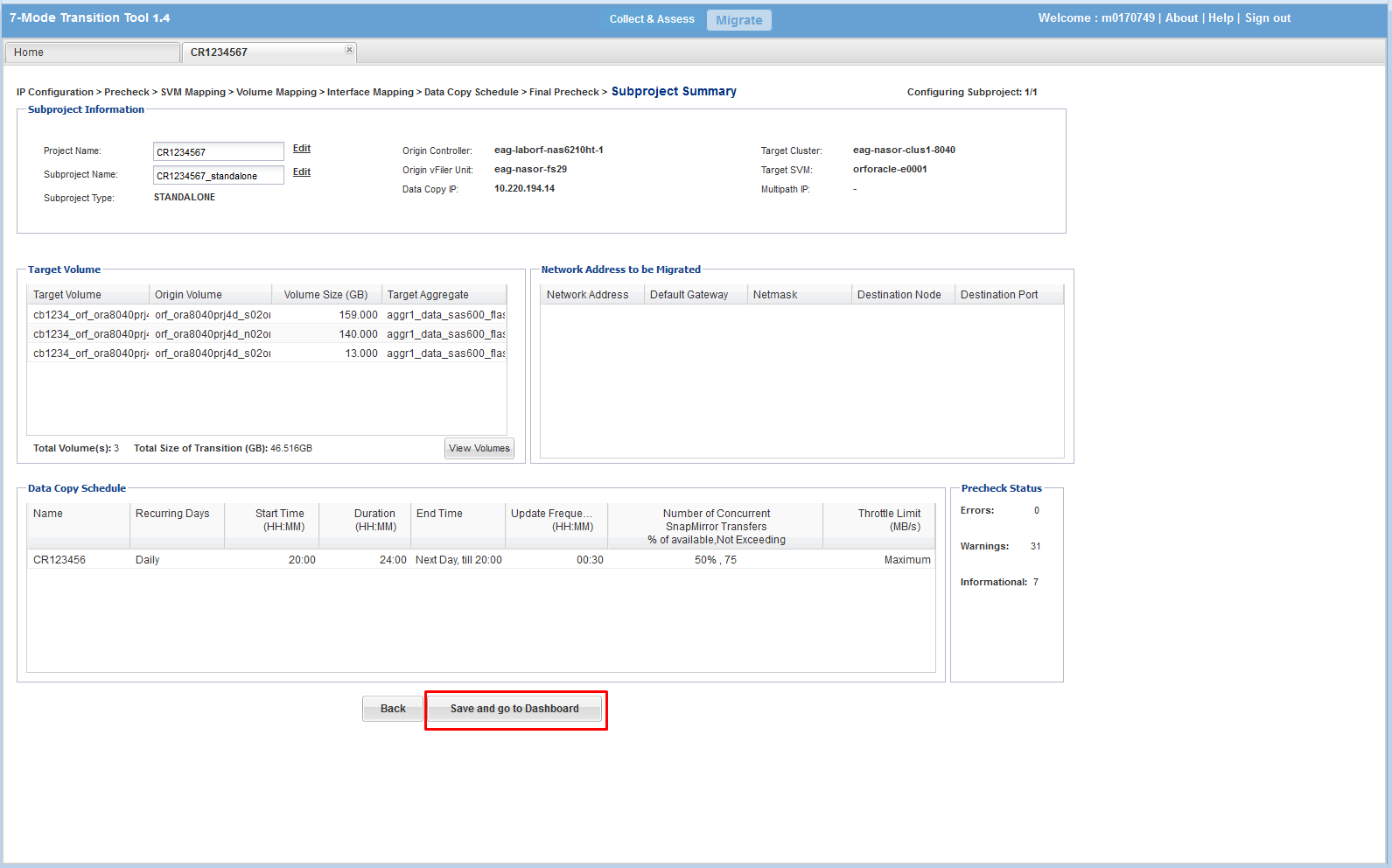
* The newly created schedule will appear
* CLICK ‘Next’



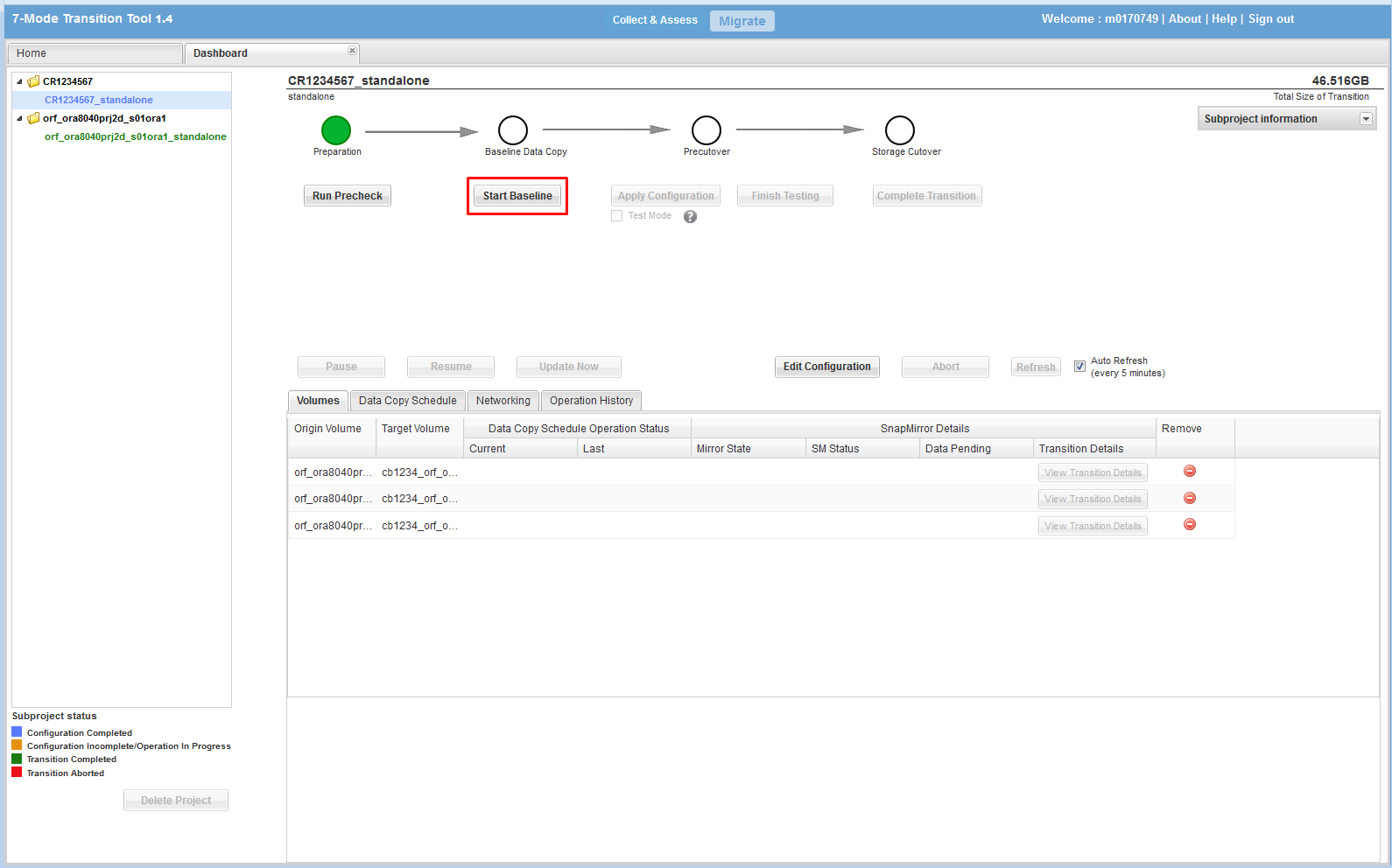
* Run PreCheck and make sure there are no errors
* CLICK ‘Run Precheck’; REVIEW the output
* CLICK ‘Close’
* CLICK ‘Next’



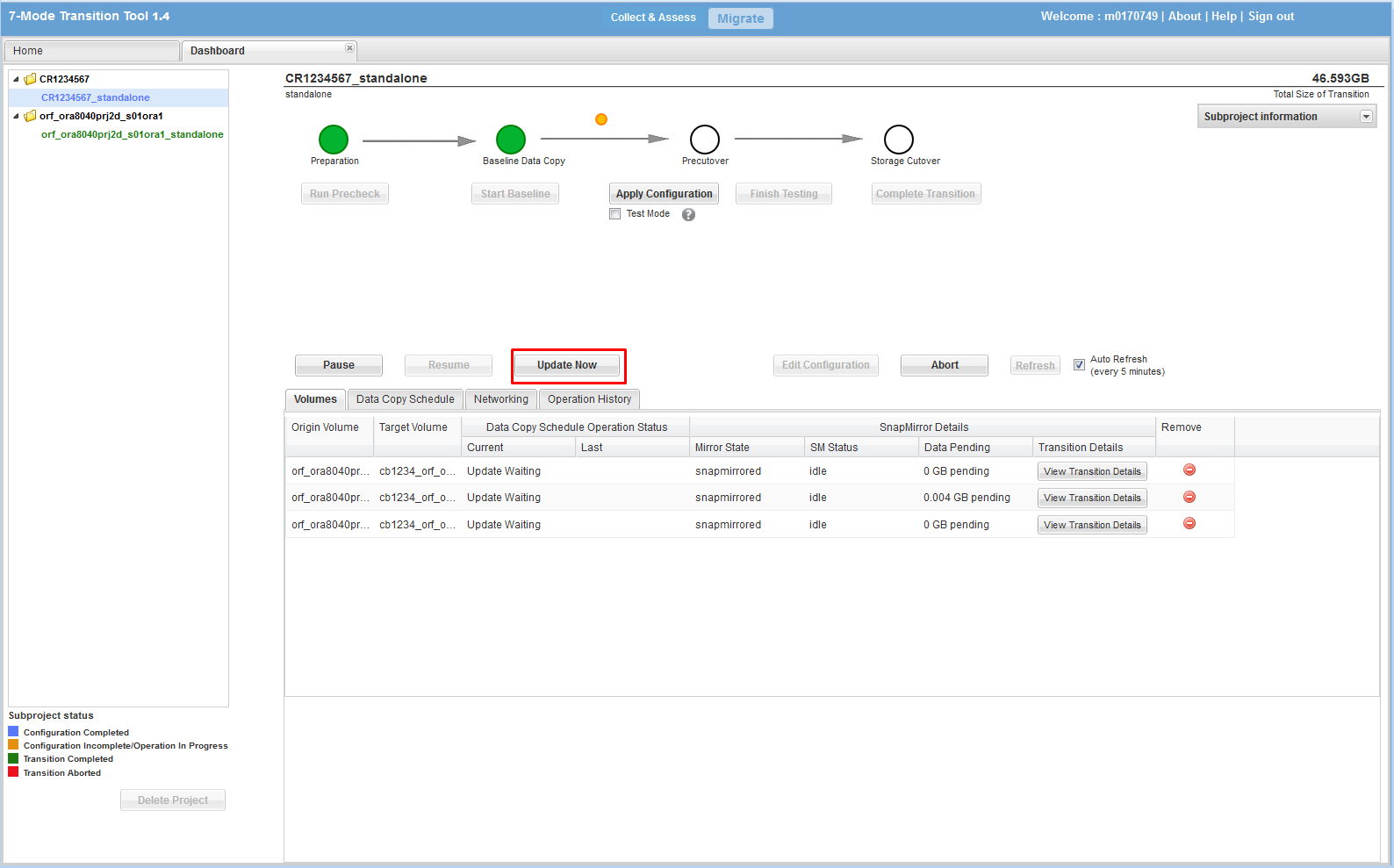
* Review the output
* CLICK ‘Save and go to Dashboard’



* A data transfer workflow will be displayed in the top of the page. The circle above ‘Baseline Data Copy’ will have a white fill color before your first transfer. It will have an orange fill color while the first transfer is running. Then a green fill color after your first successful transfer.
* CLICK ‘Run Pre-Check’; REVIEW output
* CLICK ‘Start Baseline’
* A popup window will appear, CLICK ‘yes’

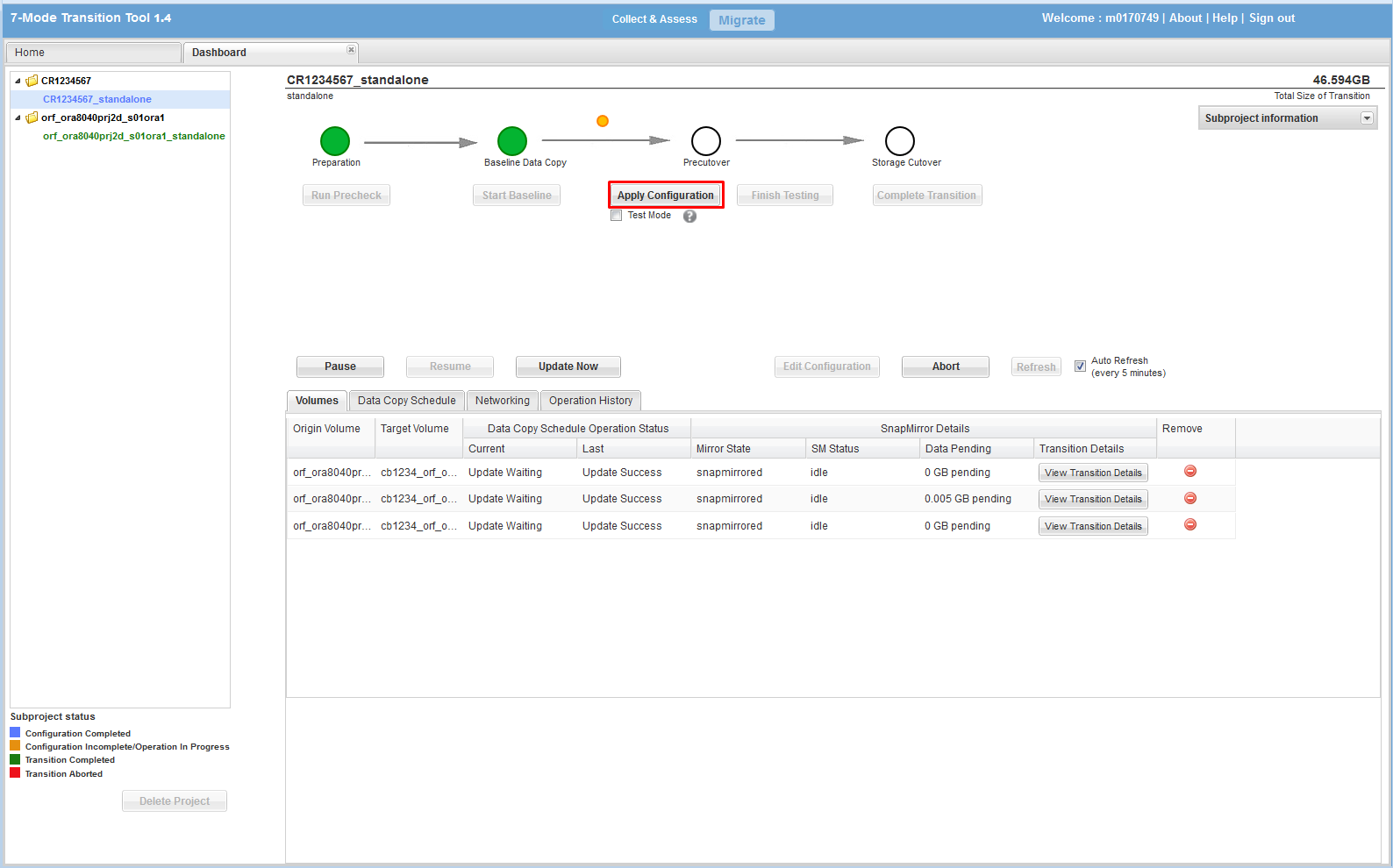


* Execute an update transfer before the cutover window so that you have minimal changes to transfer during the cutover window:
* CLICK ‘Update Now’

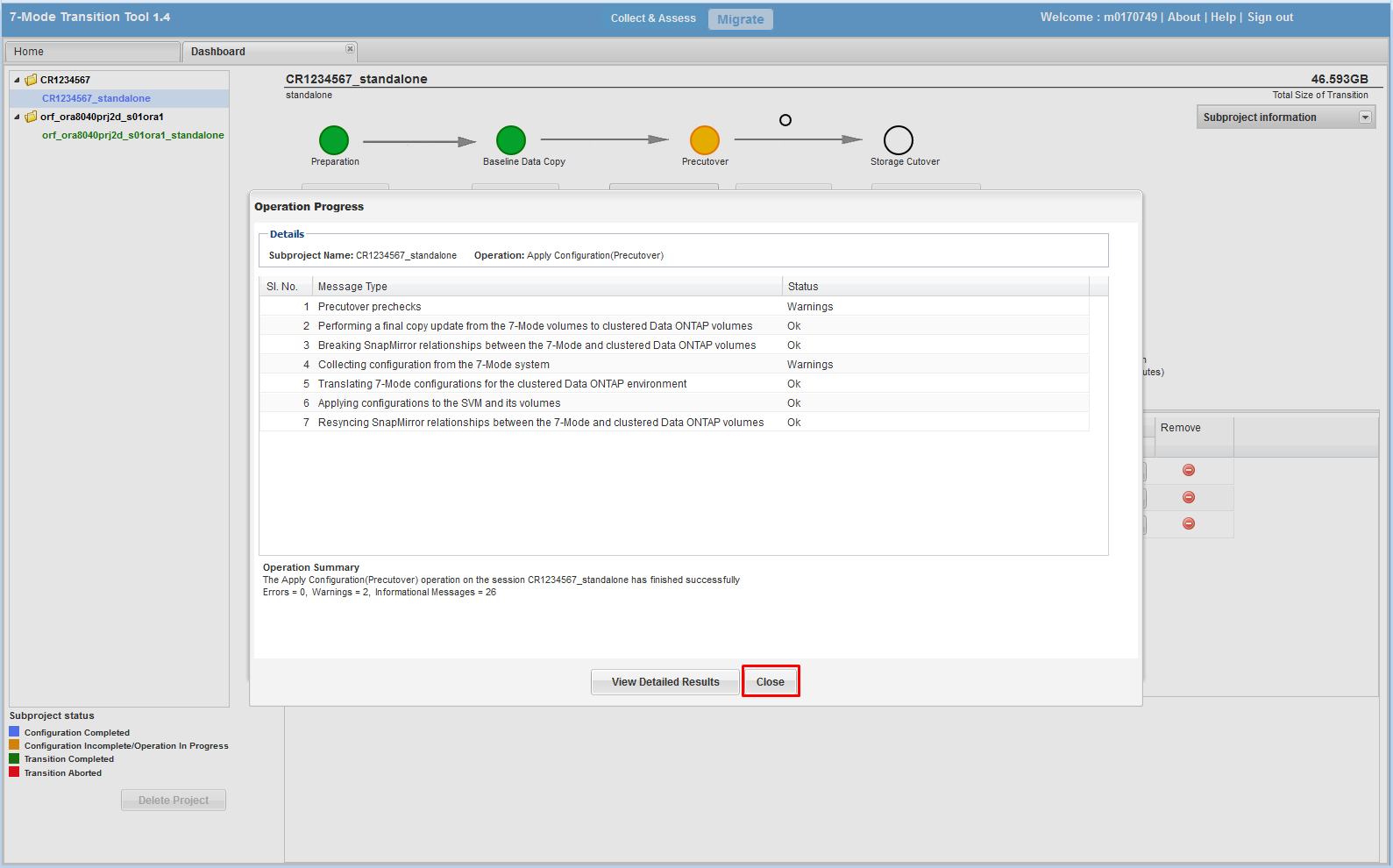


Ensure the steps below are completed during the downtime window before proceeding:

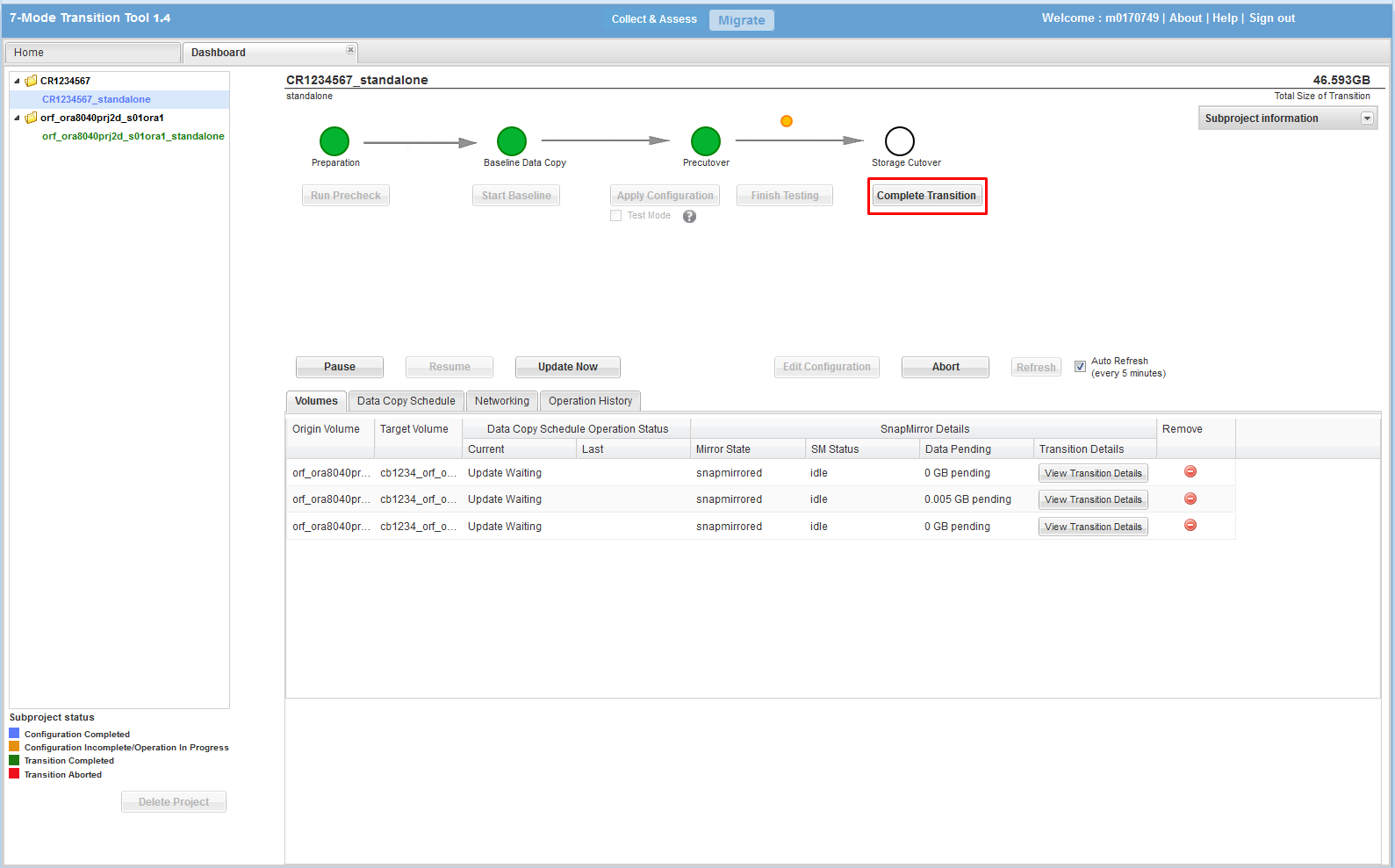
* The database has been shut down
* Source volumes have been unmounted
  + Start the cutover process:
  + CLICK ‘Apply Configuration’
  + A popup box will appear, CLICK ‘Continue’



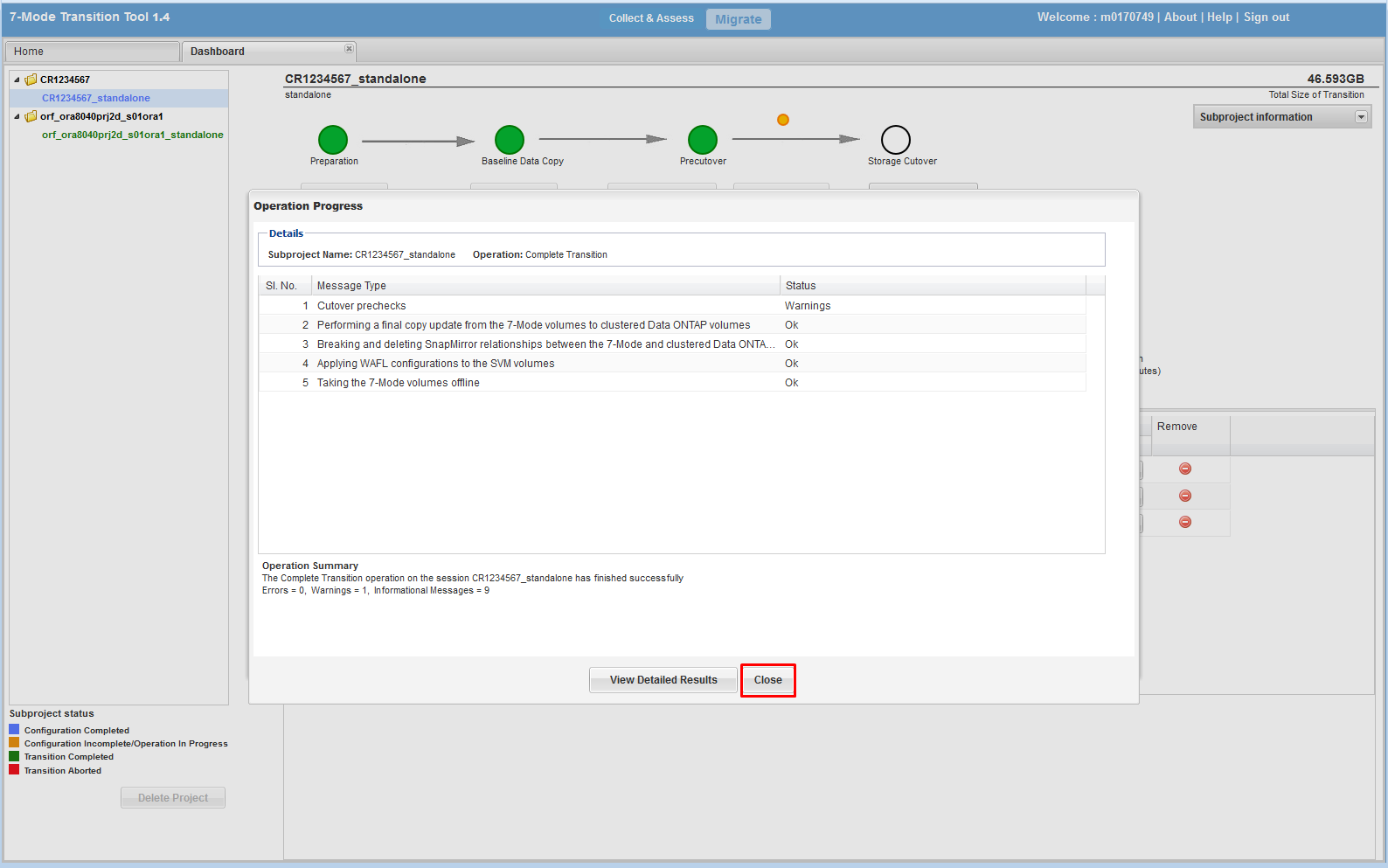
* Review log for errors
* CLICK ‘Close’



* CLICK ‘Complete Transition’
* A popup box will appear, CLICK ‘yes’
* A popup box will appear stating that the source volume will be made offline, CLICK ‘Continue’



* Review the log output for a successful transition
* CLICK ‘Close’



## ****Primary job schedule and snapshot policy configuration****

### Create job schedule and snapshot policy (only execute for OraAdmin volume):

job schedule cron create -name <volume\_name> -minute <min> -hour <hour>

volume snapshot policy create -vserver <vsname> -policy <volume\_name> -enabled true -schedule1 <volume\_name> -count1 7 -snapmirror-label1 snapvault -prefix1 sv\_<volume\_name>

job schedule cron show -name <vol\_name>

volume snapshot policy show –vserver <vsname>

## ****Volume creation****

### Create export policies (repeat this step for each volume)

vserver export-policy create –vserver <vsname> –policyname <volume\_name>

### Create the default export-policy (repeat this command for every nfs client)

vserver export-policy rule create -vserver <vsname> -policyname default

-clientmatch <nfsclient> -rorule sys -rwrule sys -superuser sys

### Create export policy rule (repeat this command for each volume and nfs client)

vserver export-policy rule create -vserver <vsname> -policyname <volume\_name> -clientmatch <nfsclient> -rorule sys -rwrule sys -superuser sys

### Add volumes to export & snapshot policies (execute for oraadmin):

volume modify -volume <volume\_name> -policy <volume\_name> -snapshot-policy <volume\_name> -vserver <vserver>

### Add volumes to export & snapshot policies (execute for \*ora\*\_nosnap & \*ora\*\_snap):

volume modify -volume <volume\_name> -policy <volume\_name> -snapshot-policy none -vserver <vserver>

volume show -vserver <vsname>

vserver export-policy rule show

### Delete snapshot policies which were created by 7MTT (names start with ‘SnapShot\_\*):

snapshot policy show -vserver <vserver>

snapshot policy delete -vserver <vserver> -policy <policy\_name>

### Rename junction paths (execute for each volume)

volume unmount -vserver <vserver> -volume <volume\_name>

volume mount -vserver <vserver> -volume <volume\_name> -junction-path /<volume\_name>

volume show -vserver <vserver> -junction

### Setup snap autodelete on volumes (run for each SNAP volume)

volume modify -vserver <vsname> -volume <volname> -space-mgmt-try-first snap\_delete

volume snapshot autodelete modify -vserver <vsname> -volume <vol\_name> -enabled true

volume snapshot autodelete modify -vserver <vsname> -volume <vol\_name> -trigger snap\_reserve

volume snapshot autodelete show -vserver <vsname>

volume show -vserver <vsname> -fields space-mgmt-try-first

volume snapshot autodelete show -vserver <vsname>

### Create qtrees quotas (repeat command below for each required qtree)

volume quota policy rule create -vserver <vsname> -policy-name default -volume <nosnap\_volname> -type tree -target <qtname> -disk-limit <XXXg>

### Enable the quotas (repeat step for each volume>

volume quota on -vserver <vsname> -volume <volname>

## ****QoS policy group creation****

### Create QoS policy group and apply it at the volume level (create one QOS policy for each volume)

qos policy-group create -policy-group <volname> -vserver <vsname> -max-throughput 6000iops

volume modify -vserver <vsname> -volume <volume> -qos-policy-group <volname>

qos policy-group show

volume show –vserver <vsname> -fields qos-policy-group

# Snapvault Configuration

## Cluster and Vserver Peering

### Confirm that cluster peering has been enabled

cluster peer show

### Create the cluster peer (skip this step if cluster peering has been configured)

cluster peer create -peer-addrs <remote\_ICL\_IP1,remote\_ICL\_IP2> -username admin

cluster peer show

### Confirm if vserver peering has been configured

vserver peer show

### Create vserver peering on the destination system (skip this step if vserver peering has been configured)

vserver peer create -vserver <destination\_vserver> -peer-cluster <source\_cluster> -peer-vserver <source\_vserver> -applications snapmirror

vserver peer show

### Accept the vserver peering on the source system

vserver peer accept -vserver <source\_vserver> -peer-vserver <destination\_vserver>

vserver peer show

## SnapVault configuration

### There are two data protection configurations for LION: SNAP & NOSNAP. The ‘oradata’ and ‘oraadmin’ volume are vaulted in a SNAP configuration. Only the ‘oraadmin’ volume is vaulted in a NOSNAP configuration. Hence, the number of volumes you execute commands for may vary based on configuration type.

### Create secondary volumes for SnapVault as type “DP” on the destination cluster

volume create -vserver <vserver> -volume <volume\_name> -aggregate <aggr\_name> -size <size> -security-style unix -space-guarantee none -percent-snapshot-space 0 -language <vol\_language> -type DP

volume show

### Create a cron job schedule if it does not exist in the destination

job schedule cron create -name xdp\_<hour> -minute 00 -hour <hour>

job schedule show

### Configure a snapmirror policy on the destination (execute these command for oraadmin & oradata)

* **7 day retention will have a snapshot count of 14 on the secondary**
* **14 day retention will have a snapshot count of 28 on the secondary**
* **30 day retention will have a snapshot count of 60 on the secondary**
* **45 day retention will have a snapshot count of 90 on the secondary**

snapmirror policy create -vserver <vserver> -policy <volume>

snapmirror policy add-rule -vserver <vserver> -policy <volume> -snapmirror-label snapvault -keep <retention#>

snapmirror show -destination-path \* -fields Schedule

snapmirror policy show

### Initialize SnapVault relationship on the destination (execute these command for oraadmin & oradata)

snapmirror create -source-path <source\_vserver>:<source\_volume> -destination-path <destination\_vserver>:<destination\_volume> -type XDP -schedule <schedule\_name> -policy <policy\_name>

snapmirror initialize -destination-path <destination\_vserver>:<destination\_volume>

snapmirror show