



Technical Report

MongoDB Database on NetApp AFF8080

Customer Blueprint

Ranga Sankar, NetApp
February 2018 | TR-4659

Abstract

This document describes the installation of MongoDB database on NetApp® AFF8080 in a VMware vSphere ESXi environment with RHEL 7.2 guests and the use of NetApp SnapCenter® for data protection.

TABLE OF CONTENTS

1	Introduction	3
2	Purpose	3
3	Solution Overview	3
4	Configuration Details	4
5	Setup Details	4
5.1	Setting Up MongoDB with ESX Guests	5
5.2	Setting Up MongoDB in RHEL 7.2 on Local Disk	5
5.3	Starting the Mongo Shards, Config Servers, and Application Servers	5
5.4	Verifying That MongoDB Is Running	6
5.5	Provisioning AFF8080 for MongoDB	6
5.6	ESX LUN Discovery and Datastores	6
5.7	Replacing Local LUNs with AFF LUNs and Restarting MongoDB	6
5.8	Restarting the MongoDB Service with the Database in AFF	6
5.9	Integrating SnapCenter with MongoDB	7
5.10	Generating MongoDB I/O to the Shards Using YCSB Client	7
6	Data Protection: Backup and Restore Using SnapCenter	8
6.1	SnapCenter Backup to Local Volume	8
7	Troubleshooting	9
	Where to Find Additional Information	10

LIST OF TABLES

Table 1)	Hardware and software details	4
----------	-------------------------------	---

LIST OF FIGURES

Figure 1)	Solution overview	4
Figure 2)	Creating backups in SnapCenter	8
Figure 3)	Thin-provisioned LUNs out of space	9
Figure 4)	Disabled datastores	10
Figure 5)	Unmounting a LUN	10

1 Introduction

MongoDB is an open-source NoSQL database used by companies of all sizes across all industries and for a wide variety of applications. These applications include business-critical operational applications for which low latency, high throughput, and continuous availability are crucial.

MongoDB is an agile database that uses a flexible document data model so that schemas can change quickly as applications evolve. This database combines the functionality that developers expect from traditional databases, such as secondary indexes, an expressive query language, and strong consistency, with the performance and agility of a NoSQL database.

MongoDB is built for scalability, performance, and high availability, scaling from single-server deployments to large, complex, multisite architectures. By leveraging in-memory computing, MongoDB provides high performance for both reads and writes. MongoDB's native replication and automated failover enable enterprise-grade reliability and operational flexibility. By deploying MongoDB on NetApp technology, you get consistent microsecond response, backup, and recovery from the cloud, helping you maintain the highest levels of performance and uptime. With the advanced fault recovery features and easy in-service growth capabilities of NetApp solutions, you can meet ever-changing business requirements.

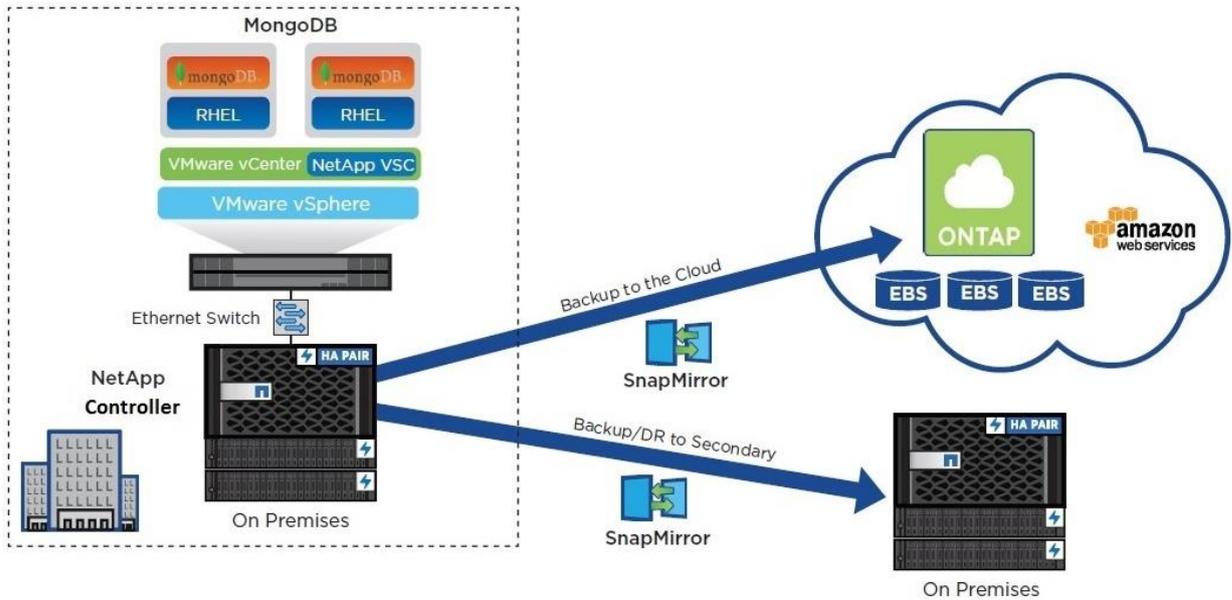
2 Purpose

The purpose of this document is to outline the steps for installing MongoDB in a VMware vSphere ESXi 6.0 environment with RHEL 7.2 guests, with storage for the shards of MongoDB from a NetApp AFF8080 array. Application-consistent backup and clone management for NetApp storage across the Data Fabric is achieved using SnapCenter in a MongoDB ecosystem. Application, system, and storage administrators and architects exploring to use MongoDB with NetApp storage will find this document useful.

3 Solution Overview

In the end-to-end solution in Figure 1, virtualized MongoDB is hosted on NetApp AFF8080 storage running VMware vSphere ESXi 6.0. The NetApp AFF array provides low latency, inline deduplication, and compression to deliver high performance. NetApp SnapCenter enables the application and the corresponding database-consistent backup. It also enables users to create zero-cost clones of the entire MongoDB ecosystem. SnapCenter provides the capability to manage remote replication of the entire MongoDB environment to ONTAP® Cloud instances that run in the cloud or to hybrid arrays. The data that is replicated to the hybrid array or to the cloud can be used for disaster recovery. For details about the architecture of MongoDB on AFF, see [TR-4492: MongoDB on the NetApp Data Fabric](#).

Figure 1) Solution overview.



4 Configuration Details

Table 1 lists the hardware and software used in this solution.

Table 1) Hardware and software details.

Component	Version
SnapCenter	3.0
Windows OS (on SnapCenter server)	Windows Server 2012 R2
ESX/VM hosts	VMware ESXi 6.0/RHEL 7.2
MongoDB	3.4.6 (two shards of three VMs, three config servers, three app servers)
PyMongo	3.4.5
MongoDB plug-in	1.0
ONTAP version	9.3x4
Controller model	AFF8080
Processors type/cache	Intel/20/130GB

5 Setup Details

The setup is divided into multiple components, with each component being independent of the others. The intent is to use VMware ESXi guests where appropriate, having a MongoDB database running on local disks as a first step. Provisioning of the AFF8080 controller can be done separately, and the LUNs from the AFF array can then be used for hosting the MongoDB database (shards). If the user does not want to use VMware ESXi for guests, the steps in this section are still applicable for regular physical hosts.

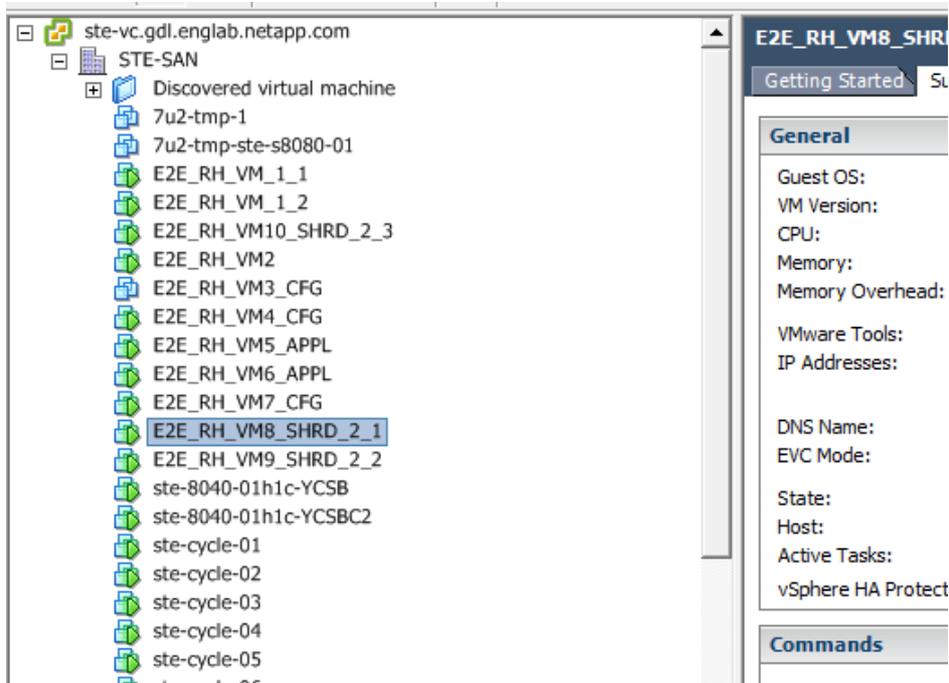
5.1 Setting Up MongoDB with ESX Guests

1. Install the ESXi bare-metal hypervisor on the local host disk. Create a vSphere environment by following the onscreen prompts of the ESXi installer.
2. Configure the ESXi host. Assign a specific IP address for the host and review the configuration options provided by the direct console.
3. Install the vSphere Client. To deploy virtual machines (VMs) and perform administrative tasks, you must install the vSphere Client and use it to manage the host. You can download the vSphere Client installer binary from your host.
4. Log in to the ESXi host and deploy the 13 VMs by using the vSphere Client.
5. Install RHEL 7.2 on 12 VMs and Windows Server 2012 R2 on one VM. Allocate three VMs for one (MongoDB) shard and another three VMs for a second shard. Allocate three VMs to be MongoDB application servers and three VMs to be MongoDB config servers. Alternatively, you can have one shard and/or one application server.

5.2 Setting Up MongoDB in RHEL 7.2 on Local Disk

1. Using `yum`, install the `mongod-org` packages version 3.4.6 or later.
2. Run the following commands to add the accounts:

```
groupadd -g 8000 MongoDB
useradd -u 8000 -g 8000 -d /MongoDBMongoDB
mkdir -p /data/db
chown -R MongoDB:MongoDB /data
chown -R MongoDB:MongoDB/data/db
```



5.3 Starting the Mongo Shards, Config Servers, and Application Servers

1. Start the `mongod` daemons on the three VMs that are allocated for the shards.

```
mongod --replset shard1/e2evm_11:27018, e2evm_12:27018, e2evm_13:27018 --shardsvr --dbpath /data/db -fork -logpath /data/db
```

2. Start the `mongod` daemons on the three VMs that are allocated as configuration servers.

```
mongod --configsvr --replset csReplSet --dbpath /data/db --port 27019 --fork -logpath /data/db
```

3. Start the `mongod` daemons on the three VMs that are allocated as application servers.

```
mongos --configdbcsReplSet/ e2ecfg_11:27019, e2ecfg_12:27019, e2ecfg_13:27019--fork -port 27017 --logpath /data/db
```

5.4 Verifying That MongoDB Is Running

1. In the RHEL VMs, execute `ps -aef | grep "mongo"`. Each of the VMs should list the `mongod` or the `mongos` process running in the VMs.

```
-bash-4.2$ ps -aef | grep mongos
mongod    3276      1  0 Oct02 ?        01:05:35 mongos  --configdb csReplSet/e2evm3:27019,e2evm4:27019,e2evm7:27019 --fork --port 27017 --logpath /data/db/mongod
.log
mongod    20211  3207  0 19:24 tty1    00:00:00 grep --color=auto mongos
-bash-4.2$ mongo --host e2evm5 --port 27017
MongoDB shell version v3.4.6
connecting to: mongod://e2evm5:27017/
MongoDB server version: 3.4.6
Server has startup warnings:
2017-10-02T16:52:52.387-0400 I CONTROL [main]
2017-10-02T16:52:52.387-0400 I CONTROL [main] ** WARNING: Access control is not enabled for the database.
2017-10-02T16:52:52.387-0400 I CONTROL [main] **      Read and write access to data and configuration is unrestricted.
2017-10-02T16:52:52.387-0400 I CONTROL [main]
mongos> show dbs
admin    0.000GB
config  0.002GB
ycsb    1.007GB
```

5.5 Provisioning AFF8080 for MongoDB

2. Create three volumes of 600GB each on the AFF controller.
3. Create three LUNs of 500GB each and three LUNs of 10GB each in the volumes created in step 1.
4. Create igroups and map the LUNs to the igroup, include the ESX FC initiators in the igroup.

5.6 ESX LUN Discovery and Datastores

1. From the ESX vSphere Client, rescan and discover the LUNs from the AFF array.
2. Create datastores in the LUNs, each of which is 500GB.

5.7 Replacing Local LUNs with AFF LUNs and Restarting MongoDB

1. From RHEL 7.2, configure a few VMs as shards. Use the datastore from the previous section and create a disk of 250GB.
2. From the RHEL 7.2 VM in step 1, invoke `rescan-scsi-bus.sh`. This step discovers the new 250GB disk as `/dev/sdb`.
3. Create a partition by running `fdisk /dev/sdb`. Press `n` and select the default for all options.
4. Run the following command:

```
Mkfs.ext4 /dev/sdb1
```

This command creates a new ext4 file system, which can then be used to create the shards.

5. Stop the `mongod` and `mongos` services in the RHEL VMs.

```
Mount /dev/sdb1 /data
Mkdir /data/db
chown -R mongod:mongod /data/db
```

5.8 Restarting the MongoDB Service with the Database in AFF

1. Restart the `mongod` daemons on the three VMs that are allocated for shards.

```
mongod --replset shard1/e2evm_11:27018, e2evm_12:27018,e2evm_13:27018 --shardsvr --dbpath /data/db -fork -logpath /data/db
```

2. Restart the `mongod` daemons on the three VMs that are allocated as configuration servers.

```
mongod --configsvr --replset csReplSet --dbpath /data/db --port 27019 --fork -logpath /data/db
```

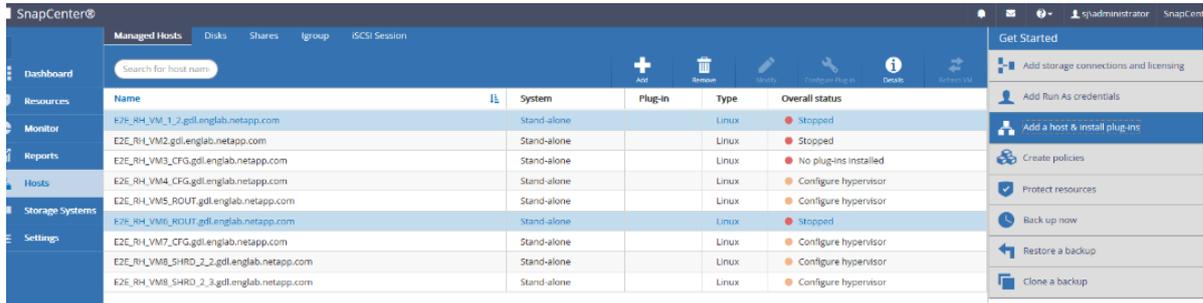
3. Restart the `mongod` daemons on the three VMs that are allocated as application servers.

```
mongos --configdbcsReplSet/ e2ecfg_11:27019, e2ecfg_12:27019, e2ecfg_13:27019--fork -port 27017 --logpath /data/db
```

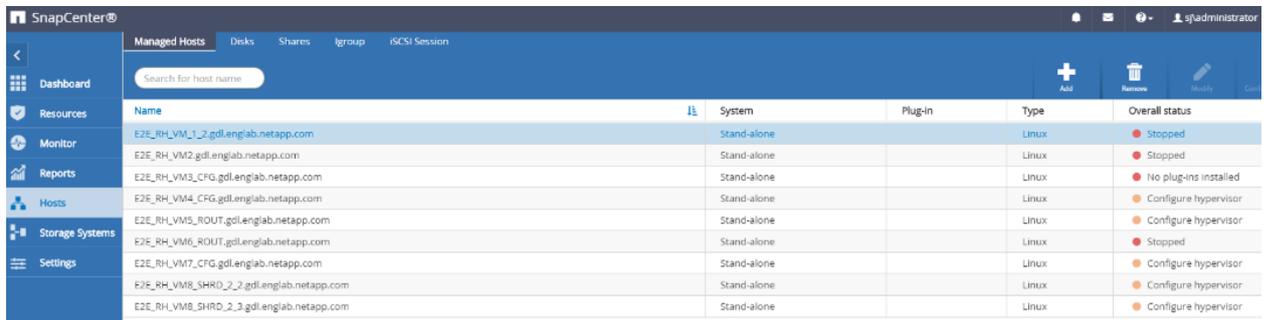
5.9 Integrating SnapCenter with MongoDB

To add the MongoDB 1.0 plug-in in SnapCenter, complete the following steps:

1. Download SnapCenter 3.0 or later from the [NetApp Support](#) site.
2. Add the ESX guests and choose the MongoDB 1.0 plug-in to install.



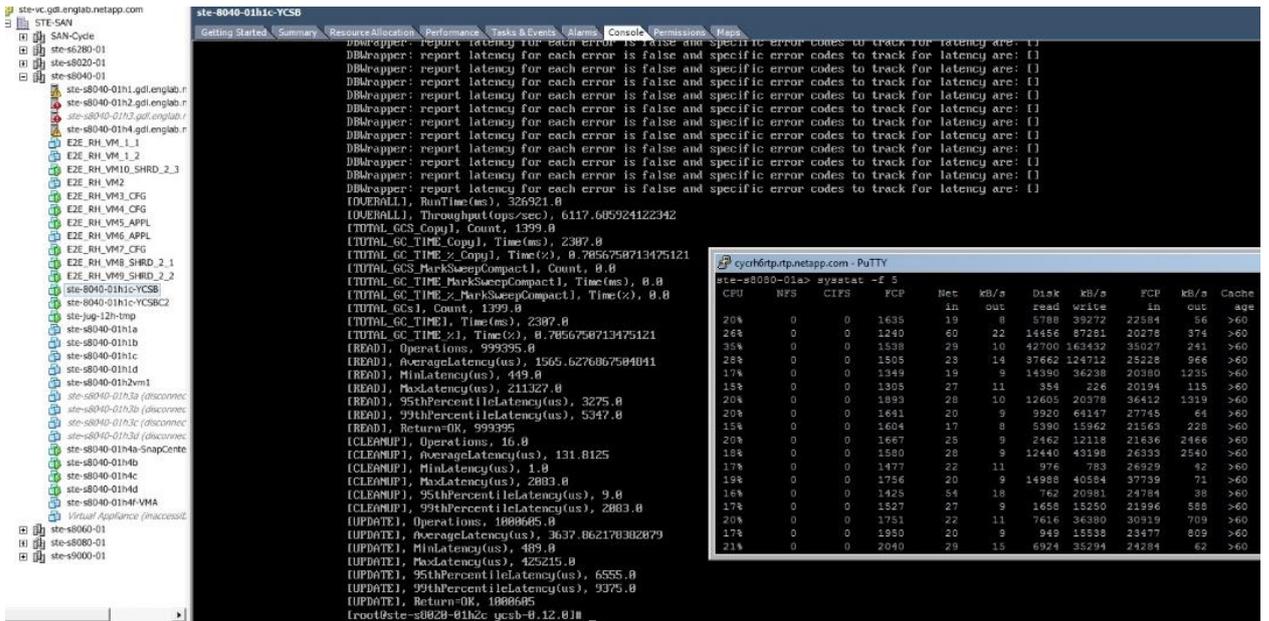
3. Install PyMongo on the ESX VM application server guests.
 - a. Obtain the whl format of PyMongo to your home directory (or example, /u/ranga/MongoDB-plugin).
 - b. Pip install ./ pymongo-3.5.1-cp26-cp26m-manylinux1_x86_64.whl.



5.10 Generating MongoDB I/O to the Shards Using YCSB Client

To generate MongoDB database traffic to the shards, install and run the YCSB tool as follows:

1. Download YCSB (Yahoo Cloud Serving Benchmark) to a VM (<https://github.com/brianfrankcooper/YCSB/wiki>).
2. Create a database using the YCSB tool in one of the shards.
3. Choose a specific workload and run the YCSB tool with the chosen workload against the database created previously. Run the YCSB tool for three minutes.
4. On the controller, execute `sysstat -a` on the node housing the LUNs that have the MongoDB shard.
5. After the YCSB tool completes, it provides a detailed output with details about the latency, throughput, and so on. Ascertain that these values look valid.



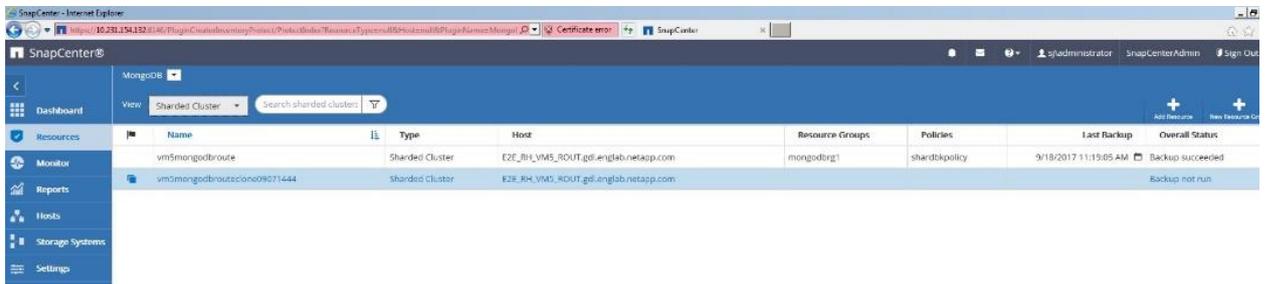
6 Data Protection: Backup and Restore Using SnapCenter

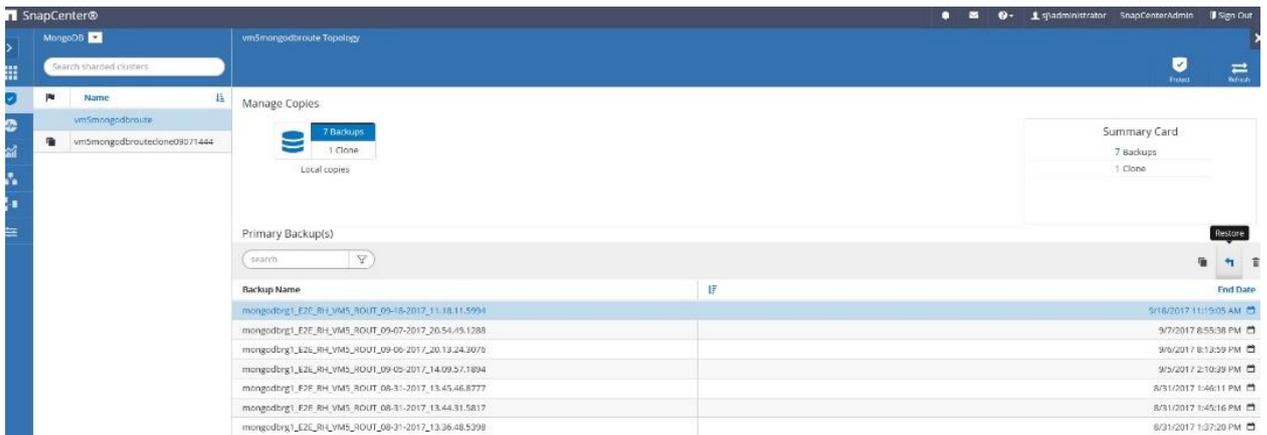
Data protection and rapid recovery are salient features offered by the Data Fabric. SnapCenter provides a single interface from which backup can be created to local volumes, to volumes in a hybrid array destination using SnapMirror, or to the cloud. Similarly, restore can be done easily from SnapCenter.

6.1 SnapCenter Backup to Local Volume

Both source and destination volumes are local to a storage virtual machine (SVM, formerly called Vserver) in this case. The backup policy for MongoDB can be easily created by using the SnapCenter menu. After the policy is created, backup can be performed on demand or during periodic intervals. The SnapCenter GUI from which backup and restore can be initiated is shown in Figure 2.

Figure 2) Creating backups in SnapCenter.

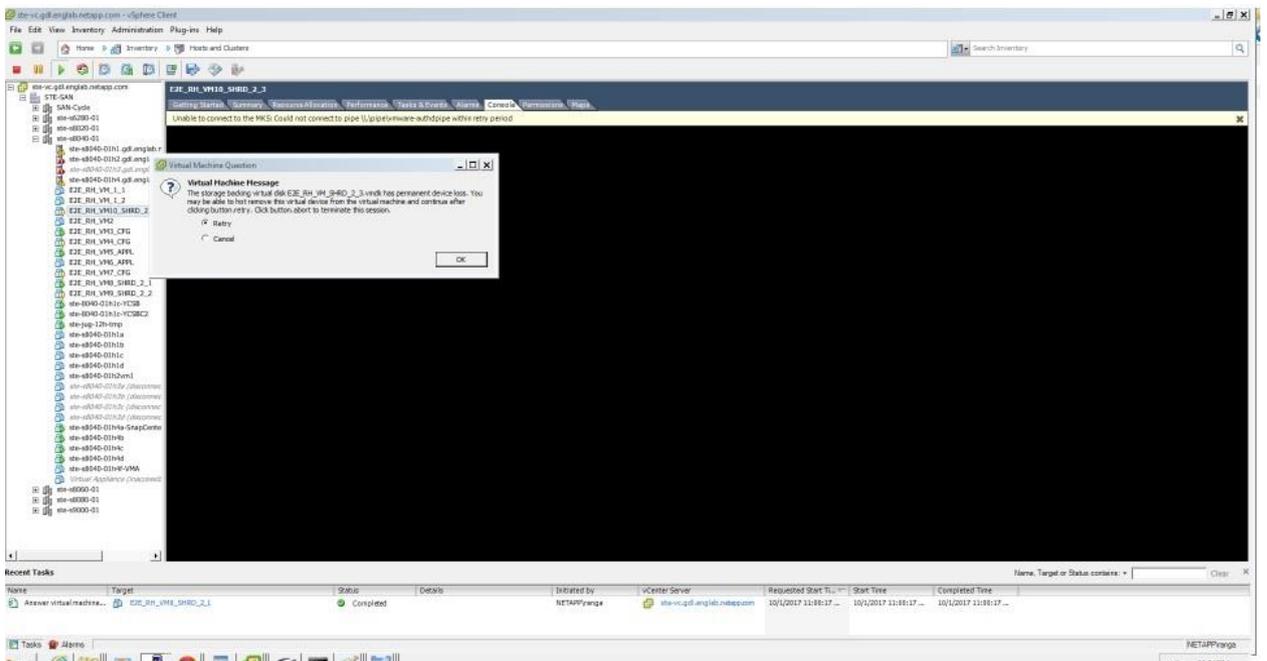




7 Troubleshooting

Issue: Thin-provisioned LUNs, containing the shard database, run out of space.

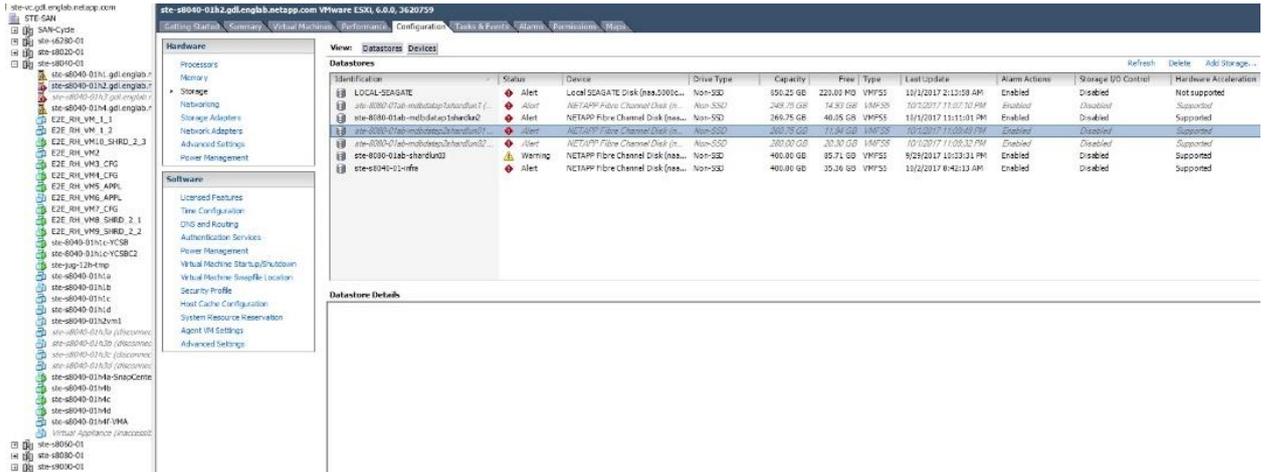
Figure 3) Thin-provisioned LUNs out of space.



Fix: For the failing VM, identify the VHDs and the corresponding datastore. Find the matching controller LUNs and bring them online if they are marked as offline. Make sure there is sufficient space. If not, increase the LUN size, rescan the LUN on the ESX host, and extend the LUN using `mkfs`.

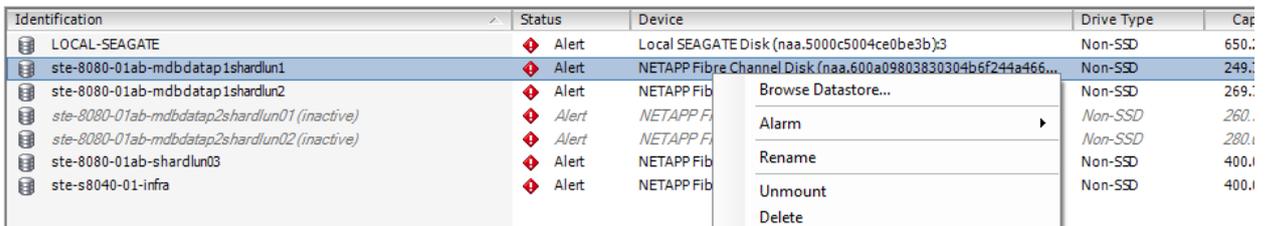
Issue: Shards run out of space, and the datastore is marked inactive.

Figure 4) Disabled datastores.



Fix: On the controller, make sure that all the LUNs are online. If the LUNs are offline, make them online. If you do not want to reboot the ESX server, find the VMs and the VHDs on the datastore that is marked active. Remove the VHDs from the VM's settings. After removing all the VHDs on the datastore, right-click the device and select Unmount. Resize the LUN on the controller, rescan on the ESX host, and do a mount. Extend the file system, and now the datastore has more space. Add the VHDs to the VMs and restart.

Figure 5) Unmounting a LUN.



Where to Find Additional Information

To learn more about the information described in this document, refer to the following documents and/or websites:

- [TR-4492: MongoDB on the NetApp Data Fabric](#)
- [NetApp Interoperability Matrix Tool \(IMT\)](#)
- [SnapCenter 3.0 Documentation Center](#)
- [TR-4660: NetApp Data Fabric for Enterprise MongoDB Solution Guide](#)

Refer to the [Interoperability Matrix Tool \(IMT\)](#) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Copyright © 2018 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

TR-4659-0218