Red Hat Reference Architecture Series

Integrating Red Hat Storage with Windows Active Directory

RHS 2.1 with Active Directory - Windows 2008 R2

Balaji Jayavelu
Principal Software Engineer

Mark Heslin
Principal Software Engineer

Version 2

October 2013

redhat.

1801 Varsity Drive™
Raleigh NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Comments and Feedback

In the spirit of open source, we invite anyone to provide feedback and comments on any reference architectures. Although we review our papers internally, sometimes issues or typographical errors are encountered. Feedback allows us to not only improve the quality of the papers we produce, but allows the reader to provide their thoughts on potential improvements and topic expansion to the papers.

Feedback on the papers can be provided by emailing refarch-feedback@redhat.com. Please refer to the title within the email.

Staying In Touch

Join us on some of the popular social media sites where we keep our audience informed on new reference architectures as well as offer related information on things we find interesting.

Like us on Facebook:

https://www.facebook.com/rhrefarch

Follow us on Twitter:

https://twitter.com/RedHatRefArch

Plus us on Google+:

https://plus.google.com/u/0/b/114152126783830728030/

Table of Contents

1 Executive Summary

With the exponential rise of Big Data solutions, increasing trend for high definition video streaming/images, cloud computing, redundant copies of regional data, regulatory requirements to store historical data etc, there is an insatiable demand for large amounts of storage. While the cost of computing has dramatically dropped, cost of conventional storage has not kept pace with it. There is a desperate need to match up with the demand and higher service level expectations, while at the same time, trying to shrink the expenditure. Large sized SAS and SATA disks are ubiquitous and 10GbE has become de-facto networking standard. There are many cost effective network based storage solutions that take advantage of these. However they fall short of expectations when it comes to scalability or flexibility.

Red Hat Storage (RHS) software offers the perfect solution to cover this gap. It is the next generation, network-oriented storage environment that leverages the power and versatility of Red Hat Enterprise Linux operating system. Red Hat Storage 2.1 provides new opportunities to unify data storage and infrastructure, increase performance, and improve availability and manageability in order to meet a broader set of an organization's storage challenges and needs. *glusterFS*, a key building block of Red Hat Storage, is based on a stackable user space design and can deliver exceptional performance for diverse workloads. glusterFS aggregates various storage servers over network interconnects into one large parallel network file system. The POSIX compatible glusterFS servers, which use XFS file system format to store data on disks, can be accessed using industry standard access protocols including NFS and SMB.

Red Hat Storage (RHS) can scale out to petabytes of data across multiple nodes serving hundreds of users. It can offer access from Windows environment via SMB (Server Message Block)/CIFS (Common Internet File System). Red Hat Enterprise Linux clients can gain access using FUSE (File System in User Space) based Native Client or NFS v3 (Network File System). Centrally authenticated access for both Windows and RHEL clients to the Red Hat Storage Servers is possible by integrating Red Hat Storage (RHS) environment with Microsoft Windows Active Directory.

The purpose of this reference architecture is to provide a self contained guide for deploying a Red Hat Storage Server that utilizes Windows Active Directory for Authentication and Access Control. While there are many implementation options available, this document focuses upon On-premise Red Hat Storage environment deployment with physical servers and local physical drives. The interoperability with Windows environment is achieved by Samba, Winbind and CTDB components which come bundled with Red Hat Storage Server software.

2 Component Overview

2.1 Red Hat Storage

Red Hat Storage (RHS) is a scale-out network attached storage (NAS) for private cloud or datacenter, public cloud, and hybrid cloud environments. It is software-only, open source, and designed to meet unstructured data storage requirements. It enables enterprises to combine large numbers of commodity storage and compute resources into a high-performance, virtualized, and centrally managed storage pool. Both capacity and performance can scale linearly and independently on-demand, from a few terabytes to petabytes and beyond, using both on-premise commodity hardware and the public cloud compute/storage infrastructure. By combining commodity economics with a scale-out approach, organizations can achieve radically better price and performance in an easily deployed and managed solution that can be configured for increasingly demanding workloads. Red Hat Storage is built using the GlusterFS open source project as the foundation.

Red Hat Storage provides the following features and capabilities:

Linear Scaling and Scale-out Architecture – Red Hat Storage is built around multiple dimension scaling model imparting flexibility to suit different demands. The capacity can be scaled up or down in a single dimension by just adding/removing one component (disks or CPU or IO) or in multiple dimensions by simultaneously adding/removing multiple components (disks or CPU or IO), achieved by adding or removing storage nodes.

Bare Metal and Cloud - Red Hat Storage Server can be deployed on bare metal machines, in the private cloud, or on a public cloud like Amazon EC2. It also possible to combine machines across public and private clouds into a storage pool.

Software-Only - Red Hat Storage is delivered as a virtual appliance, either packaged within an ISO or an image deployed in a public cloud or as a package bundle that can be deployed on an existing Red Hat Enterprise Linux server. Red Hat Storage operates in user space.

Elasticity - Red Hat Storage allows enterprises to add or delete users, application data, volumes and storage nodes without disrupting any running functionality within the infrastructure.

Decentralized with No Metadata - Unlike other storage systems with a distributed file system, Red Hat Storage does not create, store, or use a separate index of metadata in any way and instead uses Elastic Hash Algorithm, rendering a true scale out capability. The files are placed algorithmically across the different members of a volume, making them easy to retrieve and eliminating single-points-of-failure or I/ O bottlenecks. This causes performance to scale linearly when adding extra storage servers.

High Availability – Red Hat Storage supports data replication locally (N-Way Local Synchronous Replication) or over long distance (Geo-Replication asynchronous). It also supports replication in the private cloud/datacenter, public cloud or hybrid cloud environments using both N-Way and Geo-Replication methods.

Commodity Hardware - Red Hat Storage Server is designed to run on commodity x86_64 hardware. This allows one to build up an enterprise class storage network using relatively cheap components.

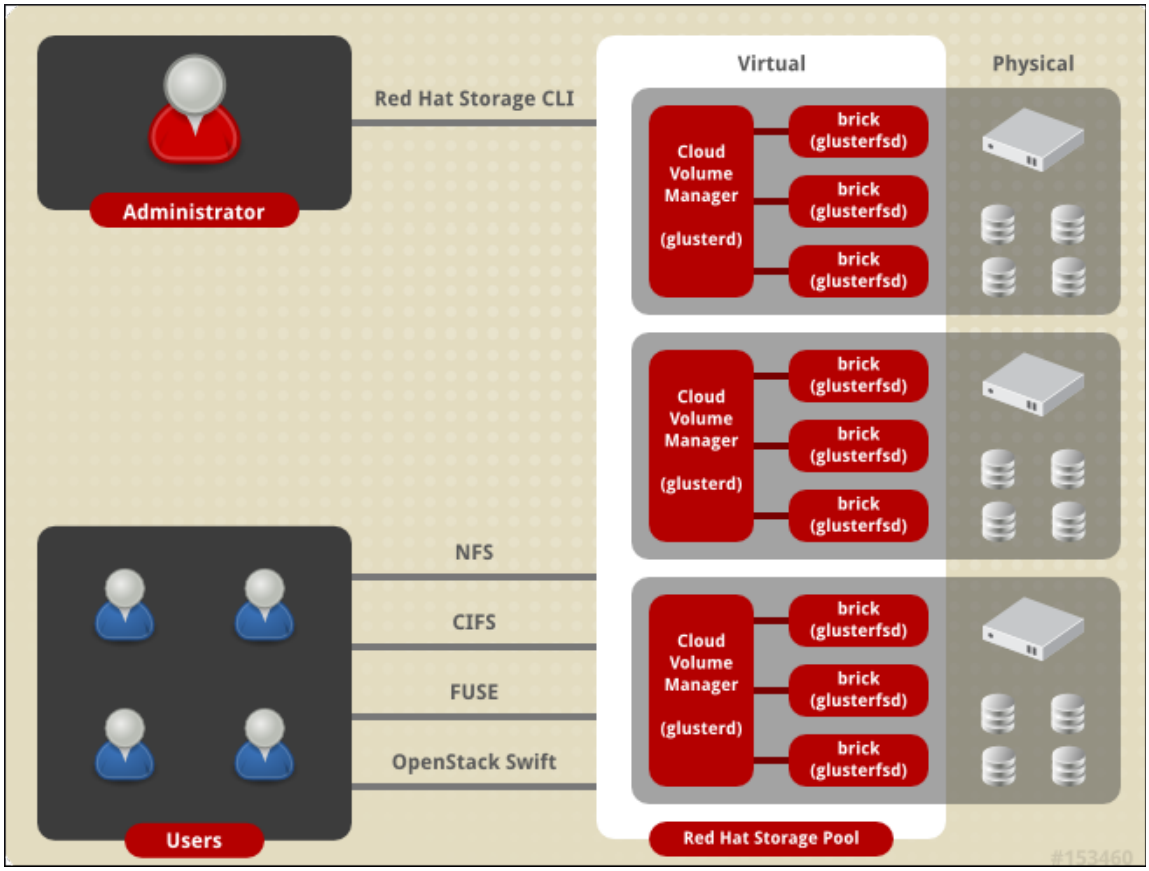


Figure 2.1-1: Red Hat Storage Architecture

2.2 Windows Server 2008 R2

Windows Server 2008 R2 is Microsoft's enterprise operating system for businesses and provides features for virtualization, power savings, manageability and mobile access. Windows Server 2008 R2 is available in several editions – Foundation, Standard, Enterprise, Datacenter, Web and HPC (High Performance Computing). Windows Server 2008 R2 Enterprise Edition is used for the configurations described in this reference architecture.

2.3 Active Directory Domain Services (AD DS)

Active Directory Domain Services is a suite of directory services developed by Microsoft. Active Directory utilizes customized versions of industry standard protocols including:

- kerberos
- Domain Name System (DNS)
- Lightweight Directory Access Protocol (LDAP)

Active Directory allows Windows system administrators to securely manage directory objects from a scalable, centralized database infrastructure. Directory objects (users, systems, groups, printers, applications) are stored in a hierarchy consisting of nodes, trees, forests and domains. Prior to Windows Server 2008 R2, Active Directory Domain Services was known as Active Directory. Active Directory Domain Services is included with Windows Server 2008 R2.

2.4 Samba

Samba is an open source suite of programs that can be installed on Red Hat Enterprise Linux 6 systems to provide file and print services to Microsoft Windows clients.

Samba provides two daemons that run on a Red Hat Enterprise Linux 6 system:

- smbd (primary daemon providing file and print services to clients via SMB)
- nmbd (NetBIOS name server - not required for integration purposes)

When combined with the reliability and simplified management capabilities of Red Hat Enterprise Linux 6, Samba is the application of choice for providing file and print sharing to Windows clients. Samba version 3.6 is used in the Samba based configurations detailed within this reference architecture.

2.4.1 CTDB (Clustered Trivial Database)

CTDB is a cluster implementation of the Trivial database (TDB) used by Samba. CTDB provides the same types of functions as TDB but in a clustered fashion, providing a TDB-style database that spans multiple physical hosts in a cluster. CTDB extends state information and inter-process communications across clustered Samba nodes in order to maintain consistent data and locking. CTDB also provides high availability (HA) features such as node monitoring, node failover and IP takeover (IPAT) in the event of a cluster node fault or failure. When a node in a cluster fails, CTDB will relocate the Internet Protocol (IP) address of the failed node to a different node to ensure that the IP addresses for the Samba file sharing services are highly available. CTDB comes bundled in with the RHS 2.1 image or along with Red Hat Enterprise Linux 6 High Availability Add-On clustering package (6.2 onwards).

2.5 SMB/CIFS

Both Server Message Block (**SMB**) and Common Internet File System (**CIFS**) are network protocols developed to facilitate client to server communications for file and print services. The SMB protocol was originally developed by IBM and later extended by Microsoft as the CIFS protocol.
Samba supports both the SMB and CIFS protocols with SMB provided for client connections to older, legacy Windows servers (Windows 2000 or earlier). The terms SMB and CIFS are often interchanged but from a functional perspective, both are protocols used by Samba.

2.6 Winbind

Winbind is a component of the Samba suite of programs that allows for unified user logon. **winbind** uses an implementation of Microsoft RPC (Remote Procedure Calls), PAM (Pluggable Authentication Modules), and Red Hat Enterprise Linux 6 nsswitch (Name Service Switch) to allow Windows Active Directory Domain Services users to appear and operate as local users on a Red Hat Enterprise Linux machine. Winbind minimizes the management of separate user accounts on both the Red Hat Enterprise Linux 6 and Windows Server 2008 R2 environments. **winbind** provides three separate functions:

- Authentication of user credentials (via PAM). This makes it possible to log onto a Red Hat Enterprise Linux 6 system (Red Hat Storage Server in this case) using Active Directory user accounts. Authentication is responsible for identifying “Who” a user claims to be.
- ID Tracking/Name Resolution via nsswitch (NSS). The nsswitch service allows user and system information to be obtained from different database services such as LDAP or NIS. ID Tracking/Name Resolution is responsible for determining “Where” user identities are found.
- ID Mapping represents the mapping between Red Hat Enterprise Linux 6 user (UID), group (GID), and Windows Server 2008 R2 security (SID) IDs. ID Mappings are handled through an idmap “backend” that is responsible for tracking “What” ID's users are known by in both operating system environments.

2.7 Kerberos

Kerberos is a network authentication protocol that uses symmetric key cryptography to provide highly secure authentication between client and server applications. Both Red Hat Enterprise Linux 6 and Windows server 2008 R2 are based on the current release of Kerberos - version 5.

Kerberos operates on the basis of “tickets” that are granted by a trusted third-party called a key distribution center (KDC). The KDC maintains a secure database of secret keys that are known only to the KDC itself and the client requesting a ticket. Tickets have a configurable expiration date and must be refreshed by the client on a regular basis.

Kerberos authentication is significantly safer than normal password-based authentication because passwords are never sent over the network - even when services are accessed on other machines.

2.8 Domain Name System (DNS)

Domain Name System (DNS) is a hierarchical, distributed naming system for managing the mappings between human-friendly domain, host and service names to IP addresses. DNS also defines the protocol for DNS communication exchanges as part of the Internet Protocol (IP) suite. On Red Hat Enterprise Linux 6, DNS client is configured in the file */etc/resolv.conf*.

2.9 Network Time Protocol (NTP)

Network Time Protocol (NTP) is an Internet protocol used to synchronize computer system clocks to a reference time source. On Red Hat Enterprise Linux 6, the **ntpd** daemon handles synchronization. NTP parameters are configured in the file */etc/ntp.conf*.

2.10 Name Service Switch (NSS)

Name Service Switch (NSS) service allows user and system information (passwd, shadow, group, hosts, etc.) to be obtained from different database services such as DNS, LDAP, NIS or local files. On Red Hat Enterprise Linux 6, NSS parameters are configured in the file */etc/nsswitch.conf*.

3 Reference Architecture Configuration

3.1 Environment

This reference architecture consists of Red Hat Storage Servers, Windows Active Directory, Red Hat Enterprise Linux (RHEL) and Windows Clients infrastructure.

Red Hat Storage Environment:

This is an on-premise deployment with four Red Hat Storage servers installed on identical physical hardware. Each server has 12 local disks with Raid 6 configuration, mounted with XFS filesystem and and distributed Gluster volume.

Red Hat Storage Servers– Node 1 ~ 4

Component	Details
Hostname	Ad-rhs-srv1, ad-rhs-srv2, ad-rhs-srv3 and ad-rhs-srv4
Operating System	RHS 2.1 2.6.32-358.18.1.el6.x86_64
Image source	RHS-2.1-20130907.0-RHS-x86_64-DVD1.iso
System Type	Dell PowerEdge R720xd
Processor	Dual Socket, 8C (16 cores) Intel(R) Xeon(R) E5-2670@2.67GHz
Memory	128GB
Storage	12 x 930GB SAS Internal Disk Drives + 2 internal OS disks 136GB
RAID	Hardware RAID 6 – PERC H710P Controller
Network	1GB

Table 1: Red Hat Storage Server -Node 1 & 2

Windows Active Directory Environment:

The Active Directory is installed on a Windows 2008 R2 server. This is a virtual machine hosted on a Red Hat Enterprise Virtualization environment.

Component	Details
Hostname	ad-winsrv1
Operating System	Windows Server 2008 R2
System Type	Virtual Machine
CPU	1 Virtual Core
Memory	2GB
Storage	20 GB (Internal)

Table 2: Active Directory Server

Client Environment:

There are three virtual machines hosted on a Red Hat Enterprise Virtualization setup. The first VM runs Red Hat Enterprise Linux (RHEL) 6.3, the second runs Windows 2008R2 and the third runs Windows 7.

Client 1 - Red Hat Enterprise Linux Client

Component	Details
Hostname	ad-rhel1
Operating System	RHEL 2.6.32-279.11.1.el6
System Type	Virtual Machine
CPU	1 Virtual Core
Memory	2GB
Storage	20 GB (Internal)

Table 3: RHEL Client

Client 2 - Windows Client

Component	Details
Hostname	ad-win2k8r2
Operating System	Windows Server 2008 R2
System Type	Virtual Machine
CPU	1 Virtual Core
Memory	2GB
Storage	20 GB (Internal)

Table 4: Windows 2008 R2 Client

Client 3 - Windows Client

Component	Details
Hostname	ad-win7
Operating System	Windows 7
System Type	Virtual Machine
CPU	1 Virtual Core
Memory	2GB
Storage	20 GB (Internal)

Table 5: Windows 7 Client

3.2 Software Configuration

The server details and their roles as referenced in Table 6: Server Roles and Operating System Revisions

Hostname	Role	Software	Version
ad-rhs-srv1~4	Red Hat Storage Server	RHS 2.1	2.6.32-358.18.1.el6
ad-winsrv1	Active Directory Server	Windows	2008 R2
ad-rhel1	Client	RHEL 6.3	2.6.32-279.11.1.el6
ad-win2k8r2	Client	Windows	2008 R2
ad-win7	Client	Windows	Windows 7

Table 6: Server Roles and Operating System Revisions

3.2.1 Required Packages

Product/Group	Package	Version	Release	Location
samba	samba	3.6.9	160.el6	RHS Server
samba	samba-client	3.6.9	160.el6	RHS Server
samba	samba-common	3.6.9	160.el6	RHS Server
samba	samba-winbind	3.6.9	160.el6	RHS Server
samba	samba-winbind-clients	3.6.9	160.el6	RHS Server
Krb5	krb5-workstation	1.10.3	10.el6	RHS Server
CTDB	ctdb.x86_64	1.0.114.6	1.el6	RHS Server
authconfig	authconfig-gtk	6.1.12	13.el6	RHS Server
X authority file utility	xorg-x11-xauth	1.0.2	7.1.el6	RHS Server
font	liberation-sans-fonts	1.05.1.20090721	5.el6	RHS Server
CIFS utils	Cifs-utils.x86_64	4.4	5.el6	RHEL Client

Table 7: Software Packages

3.2.2 Optional Packages

Product/Group	Package	Version	Release
Man	man.x86_64	1.6f	32.el6
oddjob helper (optional)	oddjob-mkhomedir	0:0.30	5.el6

3.2.3 Security

Red Hat Storage Server RHS2.1 (RHEL Version 2.6.32-358.18.1.el6.x86_64) does not support SELinux and is disabled by default when RHS 2.1 image is installed.

```
# getenforce
Disabled
```

Note : SELinux is planned to be supported in future versions.

4 Deploy Red Hat Storage Server Infrastructure

This section details installation of Red Hat Storage Server and creation of a distributed Gluster volume.

- Prework
- Deploy Red Hat Storage Server
- RHN Register and Update
- Synchronize Time Service
- Configure DNS
- Update Host File
- Kernel Tuning using Tuned
- Create Red Hat Storage (Gluster) Volume

4.1 Prework

Prior to server installation, a virtual disk must be configured as described in **Appendix B: Creating a Virtual Disk with Raid 6 using PERC Controller**

The virtual disk has the following parameters set:
Stripe unit – 256K
Stripe width – 10 (12 disks with RAID 6)

4.2 Installation

Appendix C: Red Hat Storage Server RHS2.1 Installation has been provided as a convenience to assist in the installation of Red Hat Storage Server. For detailed description please refer to the Red Hat Storage Documentation:
https://access.redhat.com/site/documentation/Red_Hat_Storage/

4.3 RHN Register and Update

Run the **rhn_register** command to register the system with Red Hat Network. To complete registration successfully, it is required to supply the Red Hat Network username and password. Follow the on-screen prompts to complete registration of the system.

```
# rhn_register
```

Required channels:

rhel-x86_64-server-6.4.z
rhel-x86_64-server-sfs-6.4.z
rhel-x86_64-server-6-rhs-2.1

Run yum command to update and ensure the Red Hat Storage servers are kept up-to-date with security patches and bug fixes.

```
#yum update
```

4.4 Synchronize Time Service

It is essential that the time service on each Red Hat Storage node and the Windows Active Directory server are synchronized, otherwise Kerberos authentication may fail due to clock skew. In environments where time services are not reliable, best practice is to configure the Red Hat Storage nodes to synchronize time from the Windows Server 2008 R2 server.

On each Red Hat Storage node, edit the file */etc/ntp.conf* so the time is synchronized from a known, reliable time service:

```
# Enable writing of statistics records.
#statistics clockstats cryptostats loopstats peerstats
server ntp1.xyz.redhat.com
server 10.5.26.10
```

Activate the change on each Red Hat Storage node by stopping the ***ntp*** daemon, updating the time, then starting the ***ntp*** daemon. Verify the change on both servers:

```
# service ntpd stop
Shutting down ntpd:                [  OK  ]
# ntpdate 10.16.255.2
22 Mar 20:17:00 ntpdate[14784]: adjust time server 10.16.255.2 offset
-0.002933 sec
# service ntpd start
Starting ntpd:                     [  OK  ]
Configure the ntpd daemon to start on server boot:
# chkconfig ntpd on
# chkconfig --list ntpd
smb             0:off   1:off   2:on   3:on   4:on   5:on   6:off
```

4.5 Configure DNS

Edit the file */etc/resolv.conf* on each Red Hat Storage node so that the domain name and search list are specified using the fully qualified domain name (FQDN). The nameserver IP addresses should be listed in preferred lookup order :

```
domain cloud.lab.eng.bos.redhat.com
search cloud.lab.eng.bos.redhat.com
nameserver 10.nn.nnn.247                # Primary Name Server
nameserver 10.nn.nnn.248                # Alternate server 1
nameserver 10.nn.nnn.2                  # Alternate server 2
```

Similarly, the hostname on each Red Hat Storage node should be set to the FQDN. Edit the file */etc/sysconfig/network* and set the hostname to use the FQDN:

```
NETWORKING=yes
HOSTNAME=ad-rhs-srv1.cloud.lab.eng.bos.redhat.com
```

Verify on each Red Hat Storage node by running the hostname utility:

```
# hostname
ad-rhs-srv1.cloud.lab.eng.bos.redhat.com
```

4.6 Kernel Tuning using Tuned

On each node, perform kernel tuning using the tuned profile *rhs-high-throughput* as follows:

```
# tuned-adm profile rhs-high-throughput
Stopping tuned:                                            [  OK  ]
Switching to profile 'rhs-high-throughput'
Applying ktune sysctl settings:
/etc/ktune.d/tunedadm.conf:                                [  OK  ]
Calling '/etc/ktune.d/tunedadm.sh start': setting readahead to 65536 on
brick devices:                                             [  OK  ]
Applying sysctl settings from /etc/sysctl.conf
Applying deadline elevator: dm-0 dm-1 dm-2 sda sdb sdc sdd [  OK  ]
Starting tuned:                                            [  OK  ]
```

This profile performs the following:

- Increases read-ahead to 64 MB.
- Changes I/O scheduler to "deadline"
- Turns off write barriers (assumes RAID controller has non-volatile writeback caching and that disks are set to - writeback caching off)

4.7 Setting up Trusted Storage Pools

Before creating a Red Hat Storage volume, a trusted storage pool must be created, consisting of the storage servers that provide bricks to a volume. The storage pool is a trusted network of Red Hat Storage servers. From the first node other peers can be added to the trusted storage pool by using the **probe** command.

The following command must be executed from the first node *ad-rsh-srv1* to probe additional nodes that are to be added.

```
# gluster peer probe ad-rhs-srv2
peer probe: success.
# gluster peer probe ad-rhs-srv3
peer probe: success.
# gluster peer probe ad-rhs-srv4
peer probe: success.
```

Verify trusted pool

```
# gluster peer status
Number of Peers: 3

Hostname: ad-rhs-srv2
Uuid: 7b50a4cf-6786-4971-b45e-bad4a910b514
State: Peer in Cluster (Connected)

Hostname: ad-rhs-srv3
Uuid: 34437f9b-ded9-4d1f-a83d-175e587da6f3
State: Peer in Cluster (Connected)

Hostname: ad-rhs-srv4
Uuid: 7fff9465-3987-4e5a-a711-b0a7eb915d55
State: Peer in Cluster (Connected)
```

Note: While in this case the probe command is being executed from the first node, it could be issued from any node in the network that has RHS software installed. However while adding an additional node to an existing pool, the probe command has to originate from one of the trusted nodes in the pool. This ensures integrity where the request to join has to come from one of the trusted nodes only.

4.8 Creating and mounting an XFS filesystem for bricks

In this section, a Raid 6 virtual disk as described under 4.1Prework must be used to create a filesystem. The steps include identifying the LUN device path */dev/sdb* using **fdisk** command (not described here) followed by initiailzation, volume group and logical volume creation and subsequent mounting of an XFS filesystem. These steps must be performed on all nodes.

1. Initialize the disk using **pvcreate** command

```
# pvcreate --dataalignment 2560k /dev/sdb
  Writing physical volume data to disk "/dev/sdb"
  Physical volume "/dev/sdb" successfully created
```

The following command confirms the location of the first Physical Extent of this physical volume ***/dev/sdb***. It is a multiple of the requested dataalignment - 2560k, calculated out of 256k stripe unit with a stripe width of 10.

```
# pvs -o +pe_start --units k
  PV         VG           Fmt  Attr PSize          PFree          1st PE
  /dev/sda2  vg_adrhssrv1 lvm2 a--   487333888.00k             0k 1024.00k
  /dev/sdb                lvm2 a--  9762242560.00k 9762242560.00k 2560.00k
```

2. Create a volume group ***datavg*** using the LUN /dev/sdb.

```
# vgcreate datavg /dev/sdb
  Volume group "datavg" successfully created
  Using volume group(s) on command line
  Finding volume group "datavg"
```

Confirm Volume group settings using **vgdisplay** command.

```
# vgdisplay -v datavg
  --- Volume group ---
  VG Name               datavg
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur PV                1
  Act PV                1
  VG Size               9.09 TiB
  PE Size               4.00 MiB
  Total PE              148959
  Alloc PE / Size       0 / 0
  Free  PE / Size       148959 / 9.09 TiB
  VG UUID               Onvn18-zupy-RT96-83yF-Ya0A-V15j-wtuGwg

  --- Physical volumes ---
  PV Name               /dev/sdb
  PV UUID               Ayrh0b-GRuw-28R5-bLv7-7g0r-ltSd-4z11CI
  PV Status             allocatable
  Total PE / Free PE    148959 / 148959
```

3. Create a logical volume of a desired size using lvcreate command. It is preferable to allocate some free space in the volume group for additional logical volumes that can be used for CTDB (covered later in this document) and other requirements.

```
# lvcreate -l 85%FREE -n rhsdata_lvol1 datavg
  Logical volume "rhsdata_lvol1" created
```

Display the newly created logical volume

```
# lvs  | egrep 'LV|rhsdata'
  LV           VG       Attr    LSize Pool Origin Data% Move Log Copy%
  rhsdata_lvol1 datavg -wi-ao-- 7.73t
```

4. Use the mkfs.xfs command to create an XFS file system on this new logical volume.
    - Inode size = 512 bytes
    - Stripe unit = 256K
    - Stripe width = 10

```
# mkfs.xfs -i size=512 -n size=8192 -d su=256k,sw=10
/dev/mapper/datavg-rhsdata_lvol1
meta-data=/dev/mapper/data_vg-rhsdata_lvol1 isize=512    agcount=32,
agsize=64825536 blks
         =                       sectsz=512   attr=2, projid32bit=0
data     =                       bsize=4096   blocks=2074417152, imaxpct=5
         =                       sunit=64     swidth=640 blks
naming   =version 2              bsize=8192   ascii-ci=0
log      =internal               bsize=4096   blocks=521728, version=2
         =                       sectsz=512   sunit=64 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

5. Mount the XFS filesystem with the following mount options:
    - inode64
    - noatime

```
# mkdir -p /rhs/storage1
# echo "/dev/mapper/datavg-rhsdata_lvol1  /rhs/storage1 xfs inode64,acl,
noatime 1 3” >> /etc/fstab
#
#mount -a
```

Verify the filesystem and mount options

```
# mount | grep storage
/dev/mapper/datavg-adfile_lvol1 on /rhs/storage1 type xfs
(rw,noatime,inode64,acl)
```

Verify the XFS filesystem for available disk space:

```
# df -h | egrep ‘FILESYSTEM|datavg’
Filesystem                      Size   Used   Avail  Use%   Mounted on
/dev/mapper/datavg-rhsdata_lvol1 7.8T   34M   7.8T   1%     /rhs/storage1
```

Note: The mount option “**acl**” is added with the assumption that Access Control List (ACL) might be required to set permissions. This mount option can be ignored if ACL is not expected to be used.

(repeat above steps on each node)

4.9 Creating a Distributed Replicate Gluster Volume

1. Create a brick directory under /rhs/storage1. It is ideal to have a unique brick name on each server. For example *ad-rhs-srv1* has directory dfilevol01_b1, server *ad-rhs-srv2* has dfilevol01_b2 etc.

```
# mkdir /rhs/storage1/dfilevol01_b1 (on node ad-rhs-srv1)
```

   Repeat this step on each node as mentioned above.

2. Perform the following commands only on the first node *ad-rhs-srv1*

```
# gluster volume create dfilevol01  replica 2 transport tcp \
ad-rhs-srv1:/rhs/storage1/dfilevol01_b1 \
ad-rhs-srv2:/rhs/storage1/dfilevol01_b2 \
ad-rhs-srv3:/rhs/storage1/dfilevol01_b3 \
ad-rhs-srv4:/rhs/storage1/dfilevol01_b4
Creation of volume dfilevol01 has been successful. Please start the volume
to access data.
```

Note: There are different ways to combine bricks into a specific type of volume, and further these volume types can be combined in different ways. This imparts flexibility to configure volumes with performance and redundancy characteristics to match different workload scenarios. In this case, a distributed replicated volume was created. Please refer to documentation for further details in this regard: https://access.redhat.com/site/documentation/en-US/Red_Hat_Storage/2.1/html/Administration_Guide/chap-User_Guide-Setting_Volumes.html

Verify the new gluster volume:

```
# gluster volume status  dfilevol01
Volume dfilevol01 is not started
```

```
# gluster volume info  dfilevol01
Volume Name: dfilevol01
Type: Distributed-Replicate
Volume ID: 8620c046-52ee-426c-9909-674a190ed5b8
Status: Created
Number of Bricks: 2 x 2 = 4
Transport-type: tcp
Bricks:
Brick1: ad-rhs-srv1:/rhs/storage1/dfilevol01_b1
Brick2: ad-rhs-srv2:/rhs/storage1/dfilevol01_b2
Brick3: ad-rhs-srv3:/rhs/storage1/dfilevol01_b3
Brick4: ad-rhs-srv4:/rhs/storage1/dfilevol01_b4
```

3. Start gluster volume

```
# gluster volume start  dfilevol01
Starting volume dfilevol01 has been successful
```

Verify the new gluster volume:

```
# gluster volume status  dfilevol01
Status of volume: dfilevol01
Gluster process                                         Port   Online    Pid
------------------------------------------------------------------------------
----
Brick ad-rhs-srv1:/rhs/storage1/dfilevol01_b1     49154     Y      22650
Brick ad-rhs-srv2:/rhs/storage1/dfilevol01_b2     49154     Y      4550
Brick ad-rhs-srv3:/rhs/storage1/dfilevol01_b3     49152     Y      3093
Brick ad-rhs-srv4:/rhs/storage1/dfilevol01_b4     49152     Y      2397
NFS Server on localhost                           2049      Y      16923
Self-heal Daemon on localhost                     N/A       Y      16930
NFS Server on ad-rhs-srv2                         2049      Y      19482
Self-heal Daemon on ad-rhs-srv2                   N/A       Y      19489
NFS Server on ad-rhs-srv3                         2049      Y      5675
Self-heal Daemon on ad-rhs-srv3                   N/A       Y      5682
NFS Server on ad-rhs-srv4                         2049      Y      5113
Self-heal Daemon on ad-rhs-srv4                   N/A       Y      5120
There are no active volume tasks
```

Notice the status as started:

```
# gluster volume info  dfilevol01
Volume Name: dfilevol01
Type: Distributed-Replicate
Volume ID: c1642388-922b-4424-8f46-3245d87677d3
Status: Started
Number of Bricks: 2 x 2 = 4
Transport-type: tcp
Bricks:
Brick1: ad-rhs-srv1:/rhs/storage1/dfilevol01_b1
Brick2: ad-rhs-srv2:/rhs/storage1/dfilevol01_b2
Brick3: ad-rhs-srv3:/rhs/storage1/dfilevol01_b3
Brick4: ad-rhs-srv4:/rhs/storage1/dfilevol01_b4
```

5 Red Hat Storage – Active Directory Integration

5.1 Prerequisites

Before integration the following steps have to be completed on an existing Red Hat Storage Environment.

5.1.1 Windows 2008 R2 Server

This section describes the building of a Windows 2008 R2 server and configuring Active Directory Domain Services.

5.1.1.1 Deploy Windows 2008 Server R2

The following Microsoft TechNet article contains the most current and comprehensive details on installing and deploying Windows Server 2008 R2: http://technet.microsoft.com/en-us/library/dd283085.aspx

For this reference architecture, a Red Hat Enterprise Virtualization VM was deployed as the Windows 2008 Server R2 server. This server contains one Virtual core, 2GB memory and 20GB of disk space.

5.1.1.2 Configure Active Directory Domain Services

Appendix B:Active Directory Domain Services – Configuration has been provided as a convenience to assist in the installation and configuration of Active Directory.

5.1.2 Red Hat Storage Server update

In addition to Red Hat Storage Servers setup as described in Section **4Deploy Red Hat Storage Server Infrastructure** the following updates must be performed.

5.1.2.1 Update DNS

Proper resolution of DNS hostnames from each Red Hat Storage node and the Windows Active Directory server are essential. Improperly resolved hostnames are one of the leading causes for integration failures. In environments where DNS lookups are not reliable, best practice is to configure the Red Hat Storage nodes to perform DNS lookups from the Windows Server 2008 R2 Active Directory server.

Edit the file */etc/resolv.conf* on each Red Hat Storage node so that the domain name and search list are specified using the fully qualified domain name (FQDN). The nameserver IP addresses should be listed in preferred lookup order :

```
domain cloud.lab.eng.bos.redhat.com
search cloud.lab.eng.bos.redhat.com
nameserver 10.nn.nnn.100                  # Windows server specified here
nameserver 10.nn.nnn.247                  # Alternate server 1
nameserver 10.nn.nnn.2                    # Alternate server 2
```

5.1.2.2 Update Hosts File

On each Red Hat Storage node, edit */etc/hosts* and add an entry for the Windows Active Directory server:

```
#---------------------------------#
# Windows Active Directory Server: #
#---------------------------------#
#
10.16.136.53   ad-winsrv1  ad-winsrv1.cloud.lab.eng.bos.redhat.com
```

5.1.3 CTDB Configuration

The following are the steps for setting up CTDB on Red Hat Storage nodes :

5.1.3.1 Creating CTDB lock volume

1. Create logical volume

```
# lvcreate -L 256M -n ctdb_lvol1 datavg
  Logical volume "ctdb_lvol1" created
```

2. Create an XFS filesystem and mount it at the /rhs/ctdb directory

```
# mkfs.xfs -i size=512 /dev/datavg/ctdb_lvol1
meta-data=/dev/datavg/ctdb_lvol1 isize=512    agcount=4, agsize=16384 blks
         =                       sectsz=512   attr=2
data     =                       bsize=4096   blocks=65536, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0
log      =internal log           bsize=4096   blocks=1200, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

```
# mkdir /rhs/ctdb
# echo "/dev/datavg/ctdb_lvol1 /rhs/ctdb xfs defaults 0 2" >> /etc/fstab
# mount -a
# df -h | egrep 'Filesystem|ctdb'
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/datavg-ctdb_lvol1
                      252M   14M  239M   6% /rhs/ctdb
```

3. Create a directory on server *ad-rhs-srv1*

```
# mkdir /rhs/ctdb/ctdbmeta_b1  ( And /rhs/ctdb/ctdbmeta_b2 on ad-rhs-srv2..) etc)
```

(repeat above steps on each node)

4. Create ctdbmeta volume of type " Distributed-Replicate Volume" (performed *on ad-rhs-srv1*)

```
# gluster volume create ctdbmeta replica 2 \
ad-rhs-srv1:/rhs/ctdb/ctdbmeta_b1 \
ad-rhs-srv2:/rhs/ctdb/ctdbmeta_b2 \
ad-rhs-srv3:/rhs/ctdb/ctdbmeta_b3 \
ad-rhs-srv4:/rhs/ctdb/ctdbmeta_b4
Creation of volume ctdbmeta has been successful. Please start the volume
to access data.
```

Note: Replicated Gluster Volume for this scenario is ideal for replica count of 2. However for a replica count of 3 and above, Distributed or Distributed-Replicate Volume must be used. Please refer to the following link for more information on this:
https://access.redhat.com/site/documentation/en-US/Red_Hat_Storage/2.1/html/2.1_Release_Notes/sect-Documentation-2.0_Update_4_and_Update_5_Release_Notes-Tech_Preview-Test_Section_5.html

5. On all the storage nodes, update the “META=all” to the newly created volume name (i.e. “META=ctdbmeta”) in the two hook scripts, which are located at:

    - */var/lib/glusterd/hooks/1/start/post/S29CTDBsetup.sh*
    - */var/lib/glusterd/hooks/1/stop/pre/S29CTDB-teardown.sh*

```
#! /bin/bash
# RHS-2.0 only
# - The script mounts the 'meta-vol' on start 'event' on a known
#   directory (eg. /gluster/lock)
# - Adds the necessary configuration changes for ctdb in smb.conf and
#   restarts smb service.
# - P.S: There are other 'tasks' that need to be done outside this script
#   to get CTDB based failover up and running.

SMB_CONF=/etc/samba/smb.conf

CTDB_MNT=/gluster/lock
PROGNAME="ctdb"
OPTSPEC="volname:"
VOL=
# $META is the volume that will be used by CTDB as a shared filesystem.
# It is not desirable to use this volume for storing 'data' as well.
# META is set to 'all' (viz. a keyword and hence not a legal volume name)
# to prevent the script from running for volumes it was not intended.
# User needs to set META to the volume that serves CTDB lockfile.
META="ctdbmeta"
                          (...output truncated...)
```

6. Start the volume

```
# gluster volume start ctdbmeta
Starting volume ctdbmeta has been successful
```

Verify the volume status

```
# gluster volume status ctdbmeta
Status of volume: ctdbmeta
Gluster process                                 Port      Online     Pid
------------------------------------------------------------------------------
Brick ad-rhs-srv1:/rhs/ctdb/ctdbmeta_b1         49155        Y      18402
Brick ad-rhs-srv2:/rhs/ctdb/ctdbmeta_b2         49155        Y      20910
Brick ad-rhs-srv3:/rhs/ctdb/ctdbmeta_b3         49153        Y      7158
Brick ad-rhs-srv4:/rhs/ctdb/ctdbmeta_b4         49153        Y      6548
NFS Server on localhost                         2049         Y      18414
Self-heal Daemon on localhost                   N/A          Y      18424
NFS Server on ad-rhs-srv2                       2049         Y      20922
Self-heal Daemon on ad-rhs-srv2                 N/A          Y      20932
NFS Server on ad-rhs-srv4                       2049         Y      6561
Self-heal Daemon on ad-rhs-srv4                 N/A          Y      6571
NFS Server on ad-rhs-srv3                       2049         Y      7170
Self-heal Daemon on ad-rhs-srv3                 N/A          Y      7180

There are no active volume tasks
```

```
# gluster volume info ctdbmeta
Volume Name: ctdbmeta
Type: Distributed-Replicate
Volume ID: 35e765a3-f6c2-4590-82c8-b9163d2f194e
Status: Started
Number of Bricks: 2 x 2 = 4
Transport-type: tcp
Bricks:
Brick1: ad-rhs-srv1:/rhs/ctdb/ctdbmeta_b1
Brick2: ad-rhs-srv2:/rhs/ctdb/ctdbmeta_b2
Brick3: ad-rhs-srv3:/rhs/ctdb/ctdbmeta_b3
Brick4: ad-rhs-srv4:/rhs/ctdb/ctdbmeta_b4
```

7. Notice the new mount point '*/gluster/lock*'

```
# df -h | grep gluster
ad-rhs-srv1:/ctdbmeta        252M   14M  239M   6% /gluster/lock
```

8. Review the */etc/samba/smb.conf file* and ensure the 'ctdb' settings are in the [global] section of the */etc/samba/smb.conf* file

```
# ctdb config for glusterfs
        clustering = yes
        idmap backend = tdb2
#
[gluster-ctdbmeta]
comment = For samba share of volume ctdbmeta
vfs objects = glusterfs
glusterfs:volume = ctdbmeta
glusterfs:logfile = /var/log/samba/glusterfs-ctdbmeta.log
glusterfs:loglevel = 7
path = /
read only = no
guest ok = yes
```

NOTE: The above entries are added and '*/gluster/lock*' filesystem is mounted by gluster when the ctdbmeta volume is started.

5.1.3.2 Configuration files for CTDB

Steps 1,2 and 3 below, must be performed on all the nodes.

1. Create configuration files

```
# cd /gluster/lock
```

The next step is to create these configuration files for CTDB under this directory:

- ctdb
- public_addresses
- nodes

Contents of */gluster/lock/ctdb:*

```
CTDB_RECOVERY_LOCK=/gluster/lock/lockfile
CTDB_PUBLIC_ADDRESSES=/gluster/lock/public_addresses
CTDB_NODES=/gluster/lock/nodes
CTDB_MANAGES_SAMBA=yes
CTDB_MANAGES_WINBIND=yes
```

Contents of */gluster/lock/nodes:* (these are the server IPs)

```
10.16.141.1
10.16.141.2
10.16.141.3
10.16.141.4
```

Contents of */gluster/lock/public_addresses:* (these are the virtual IPs)

```
10.16.141.11/21 em1
10.16.141.12/21 em1
10.16.141.13/21 em1
10.16.141.14/21 em1
```

Note: em1 is the network device name used for the interface.

2. Move the original ctdb file and create a symbolic link

```
# mv /etc/sysconfig/ctdb /etc/sysconfig/ctdb.orig
# ln -s /gluster/lock/ctdb /etc/sysconfig/ctdb
# ln -s /gluster/lock/nodes /etc/ctdb/nodes
# ln -s /gluster/lock/public_addresses /etc/ctdb/public_addresses
```

3. Starting CTDB

```
# service ctdb start
```

4. Disable Samba on all nodes as CTDB is going to control samba.

```
# chkconfig smb off
```

5. Disable Winbind on all nodes as CTDB is going to control Winbind.

```
# chkconfig winbind off
```

This change simplifies the management of **Samba** and **Winbind** by automatically starting and stopping the **smbd** and **winbindd** daemons when the **ctdb** service is started or stopped.

Note: If winbind is not configured yet, step 5 has to be repeated after **5.2.5.1Configure Authentication.**

6. Verify ctdb configuration by checking the status

```
# ctdb status
Number of nodes:4
pnn:0 10.16.141.1      OK (THIS NODE)
pnn:1 10.16.141.2      OK
pnn:2 10.16.141.3      OK
pnn:3 10.16.141.4      OK
Generation:1581434984
Size:4
hash:0 lmaster:0
hash:1 lmaster:1
hash:2 lmaster:2
hash:3 lmaster:3
Recovery mode:NORMAL (0)
Recovery master:0
```

5.1.4 Install Samba/Winbind/X Windows Packages

Ensure the following packages are installed on the Red Hat Storage nodes before proceeding with the configuration:

krb5-workstation
samba-client
samba-winbind
samba-winbind-clients
samba-common

All the above come preinstalled with Red Hat Storage Software RHS 2.1 package. If not, they have to be manually installed.

It is recommended to use graphical user interface to configure **Winbind** using **system-config-authentication** command. To enable this interface, it is required to install the following packages:

```
# yum install authconfig-gtk
# yum install xorg-x11-xauth
# yum install liberation-sans-fonts
```

By default, .Xauthority file does not exist and is required to be able to bring up the graphical interface. After the above packages are installed, a log out and a reconnect using ssh with '-X' or '-Y' option generates this file.

5.1.5 Install software packages and configure Kerberos Client

Best practice is to install and configure the Kerberos client (**krb5-workstation**) to ensure Kerberos is able to properly authenticate to Active Directory on the Windows Server 2008 R2 server. This step is optional but highly recommended as it is useful for troubleshooting Kerberos authentication issues. Perform the steps below on each Red Hat Storage node.

Verify if the Kerberos client is installed:

```
# yum list installed | grep krb5
krb5-libs.x86_64         1.10.3-10.el6_4.6         @rhel-x86_64-server-6.4.z
krb5-workstation.x86_64  1.10.3-10.el6_4.6         @rhel-x86_64-server-6.4.z
```

If Kerberos has not been previously configured, modify the Kerberos configuration file (*/etc/krb5.conf*) by adding entries for the new Kerberos and Active Directory realms. Note the differences in the Kerberos [**realms**] and Active Directory [**domain_realm**] realm entries.

Create a safety copy of the Kerberos configuration file:

```
# cp -p /etc/krb5.conf /etc/krb5.conf.orig
```

Edit the file */etc/krb5.conf* as follows – changes are highlighted in bold:

```
[logging]
  default = FILE:/var/log/krb5libs.log
  kdc = FILE:/var/log/krb5kdc.log
  admin_server = FILE:/var/log/kadmind.log

[libdefaults]
  default_realm = AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
  dns_lookup_realm = false
  dns_lookup_kdc = false
  ticket_lifetime = 24h
  renew_lifetime = 7d
  forwardable = true

[realms]
  AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM  = {
    kdc = AD-WINSRV1.AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
    admin_server = AD-WINSRV1.AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
 }

[domain_realm]
  .ad-refarch.cloud.lab.eng.bos.redhat.com = AD-REFARCH.CLOUD.LAB.ENG.BOS.
REDHAT.COM
  ad-refarch.cloud.lab.eng.bos.redhat.com = AD-REFARCH.CLOUD.LAB.ENG.BOS.
REDHAT.COM
```

Under Kerberos, [**realms**] is set to the Kerberos server definitions and [**domain_realm**] defines the Active Directory server. Both are in the Active Directory **AD-REFARCH** domain.

Verify the Kerberos configuration. First, clear out any existing tickets:

```
# kdestroy
# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

Obtain a new Kerberos ticket:

```
# kinit administrator@AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
Password for administrator@REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM: ********
```

Verify if a new Kerberos ticket has been granted:

```
# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM

Valid starting     Expires            Service principal
09/13/13 14:03:14  09/14/13 00:03:18  krbtgt/AD-
REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM@AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
   renew until 09/20/13 14:03:14
```

At this point Kerberos is fully functional and the client utilities (*kinit, klist, kdestroy*) can be used for testing and verifying Kerberos functionality.

5.1.6 Install oddjob-mkhomedir *(optional)*

The ***oddjob-mkhomedir*** package is required to ensure that user home directories are created when a user performs a login the first time. This package comes pre-installed in the Red Hat Storage software package.

However, the clients connecting to the Red Hat Storage Servers typically do not login to the servers but just access the data. Hence this is not required and optional only when a user is expected to login to these servers using AD credentials.

5.1.7 Load Balancer Configuration

The four Red Hat Storage nodes *ad-rhs-srv1, ad-rhs-srv2, ad-rhs-srv3 and ad-rhs-srv4* are tied together by a loadbalancer configuration with the virtual server name *rhs-srv*. Based on the loadbalancer setting, the clients can be redirected to any of these four nodes.

In this reference architecture, all clients connect to *rhs-srv* server which is configured with round-robin DNS to automatically cycle through the transferable IP addresses by binding them to a single DNS hostname.

Zone file entries for redirection of the hostname *rhs-srv* to four virtual IPs in a round robin manner.

```
;
; RHS CTDB cluster alias
rhs-srv          IN      A       10.16.141.11
rhs-srv          IN      A       10.16.141.12
rhs-srv          IN      A       10.16.141.13
rhs-srv          IN      A       10.16.141.14
```

5.2 Integration

In this section, the tasks necessary for integrating Red Hat Storage nodes into an existing Windows Active Directory domain are detailed. It is assumed that the prerequisites as described in Section 5.1Prerequisites have been met.

5.2.1 Overview

This configuration is for environments looking to integrate one or more Red Hat Storage nodes into an Active Directory domain or forest with the capability to customize user configurations. Login access and file sharing services are provided.

5.2.2 Configuration Summary

Configuration Summary Samba/Winbind – idmap_ad		
Components		
RHS 2.1:	• Samba/Winbind	
Windows 2008 Server R2:	• Active Directory • Identity Management for UNIX (IMU)	
Authentication (pam)	• Windbind (pam_winbind)	
IP Failover	• CTDB	
ID Tracking/ Name Resolution (nss)	• Windbind (nss_winbind)	
ID Mapping (“back-end”)	• Windbind (idmap_ad)	
Configuration Files	• /etc/krb5.conf • /etc/samba/smb.conf	• /gluster/lock/ctdb • /gluster/lock/nodes • /gluster/lock/public_addresses
Advantages	• SID mappings homogeneous across multiple RHEL servers • Customizeable user configurations (shell, home directory) (configured within AD) • Centralized user account management • SFU, RFC2307 compatible mappings	
Disadvantages	• Requires additional configuration work to support a forest of AD domains or multiple domain trees • Requires additional user management tasks – user/group ID attributes must be set within AD	
Notes	• Requires the ability to modify user attributes within AD (via IMU)	

Table 8: Samba,Winbind,Kerberos and CTDB Configuration Information

5.2.3 Red Hat Storage Environment with Active Directory Integration

The following provides an overview of the systems and services utilized:

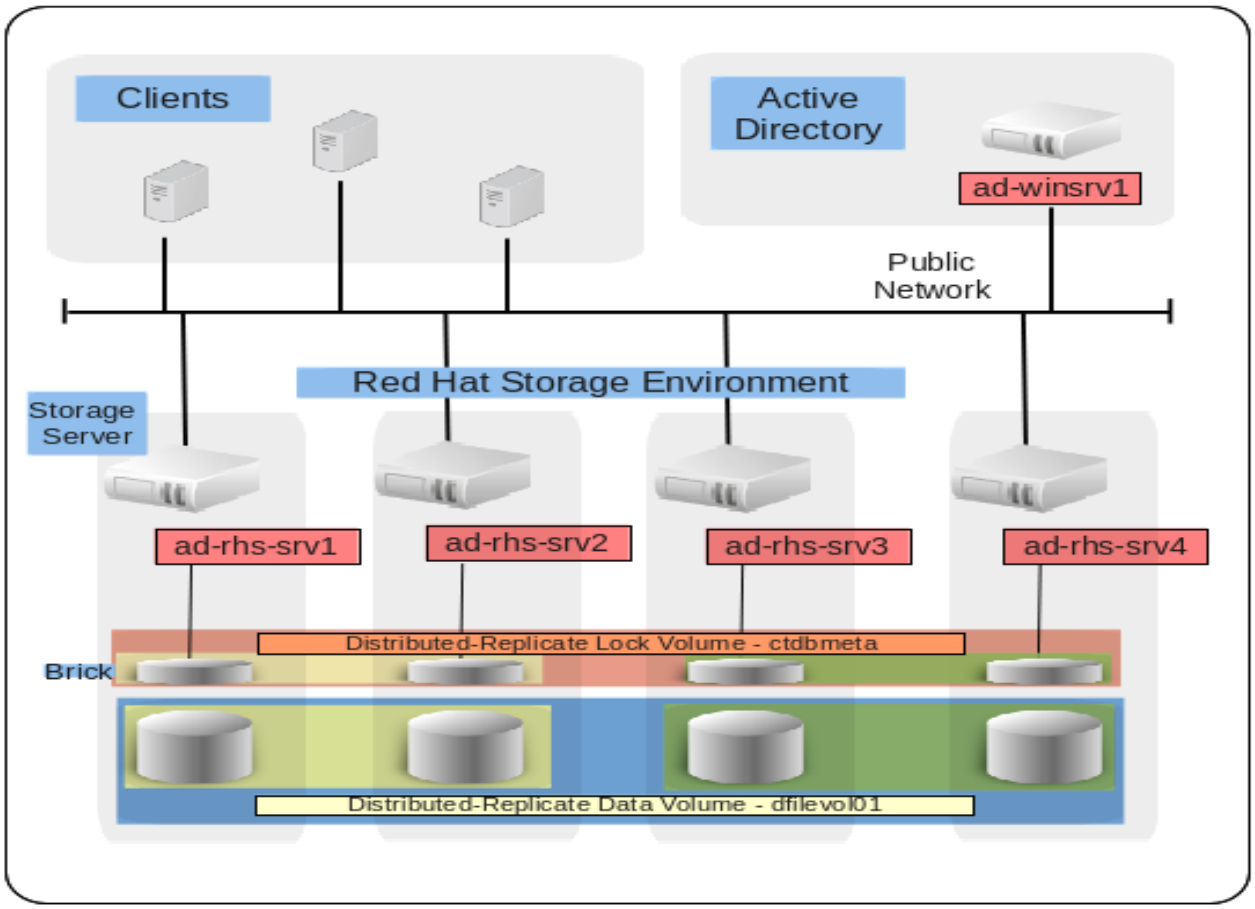


Figure 5.2.3-1: Systems Overview

5.2.4 Authentication and ID Components

The following depicts the Authentication, ID Tracking and ID Mapping:

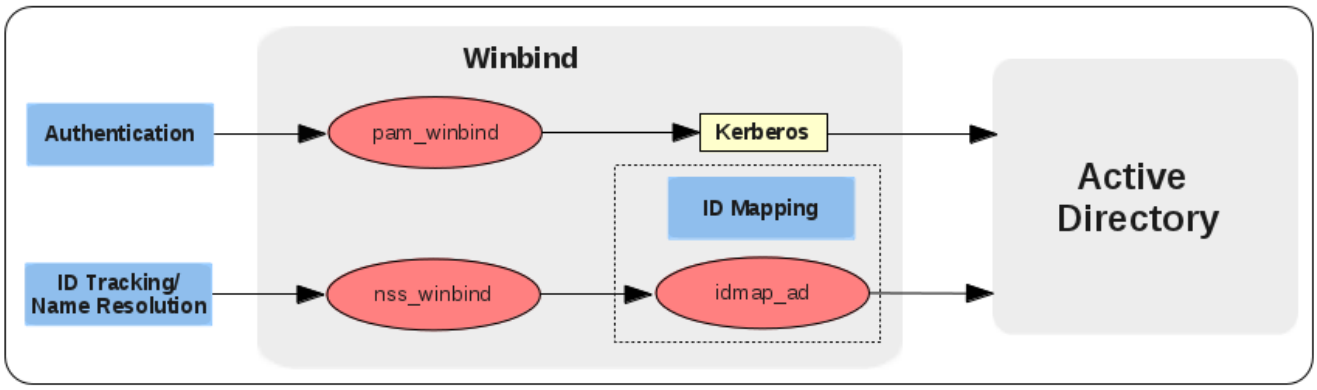


Figure 5.2.4-1: Authentication, ID Tracking and ID Mapping

5.2.5 Integration Tasks

Integrating Red Hat Storage Servers into an Active Directory domain involves the following series of steps:

1. Configure Authentication
2. Verify/Test Active Directory
3. Modify Samba Configuration
4. Verification of Services

The following provides a step-by-step guide to the integration process:

5.2.5.1 Configure Authentication

The **system-config-authentication** tool simplifies configuring the Samba, Kerberos, security and authentication files for Active Directory integration. Invoke the tool as follows:

```
# system-config-authentication
```

On the **Identity & Authentication** tab, select the **User Account Database** drop-down then select **Winbind**.

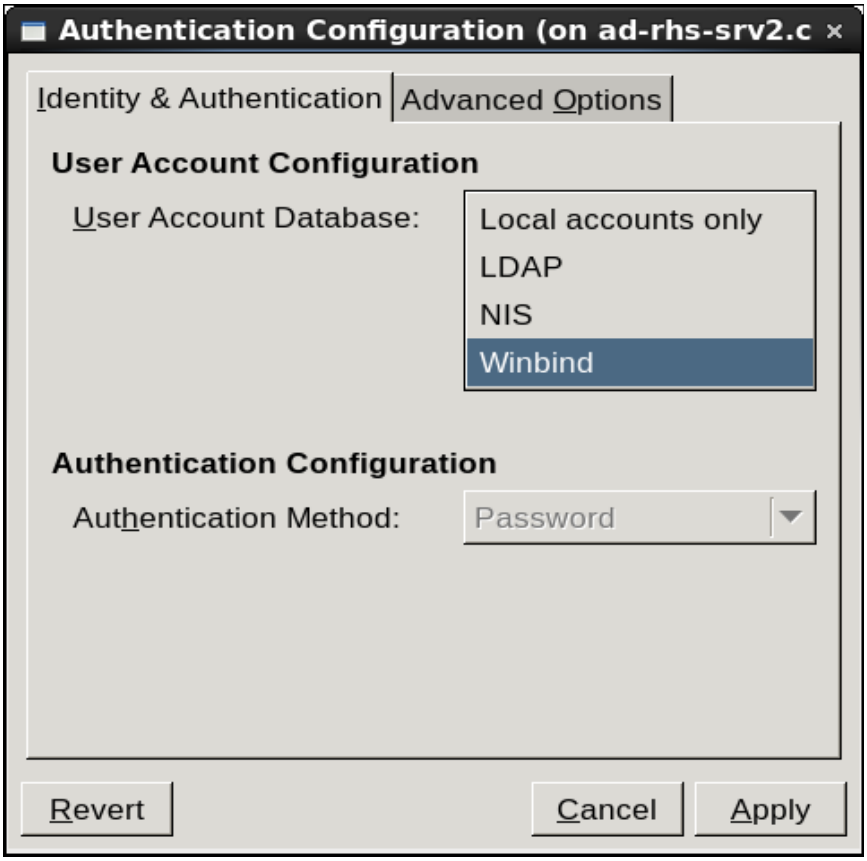


Figure 5.2.5-1: User Account Database

A new set of fields is displayed. Selecting the **Winbind** option configures the system to connect to a Windows Active Directory domain. User information from a domain can then be accessed, and the following server authentication options can be configured:

- **Winbind Domain**: Windows Active Directory domain
- **Security Model**: The Samba client mode of operation. The drop-down list allows selection of the following options:

  ads - This mode instructs Samba to act as a domain member in an Active Directory Server (ADS) realm. To operate in this mode, the krb5-server package must be installed, and Kerberos must be configured properly.

  domain - In this mode, Samba attempts to validate the username/password by authenticating it through a Windows Active Directory domain server, similar to how a Windows Server would.

  server - In this mode, Samba attempts to validate the username/password by authenticating it through another SMB server. If the attempt fails, the user mode takes effect instead.

  user - This is the default mode. With this level of security, a client must first log in with a valid username and password. Encrypted passwords can also be used in this security mode.
- **Winbind ADS Realm**: When the ***ads*** Security Model is selected, this allows you to specify the ADS Realm the Samba server should act as a domain member of.
- **Winbind Domain Controllers**: Use this option to specify which domain server winbind should use.
- **Template Shell**: When filling out the user information for a Windows user, the winbindd daemon uses the value chosen here to specify the login shell for that user.
- **Allow offline login**: By checking this option, authentication information is stored in a local cache. This information is then used when a user attempts to authenticate while offline.

Populate the fields as follows:

User Account Database:	**Winbind**
Winbind Domain:	**AD-REFARCH**
Security Model:	**ads**
Winbind ADS Realm:	**AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM**
Winbind Domain Controllers:	**AD-WINSRV1.AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM**

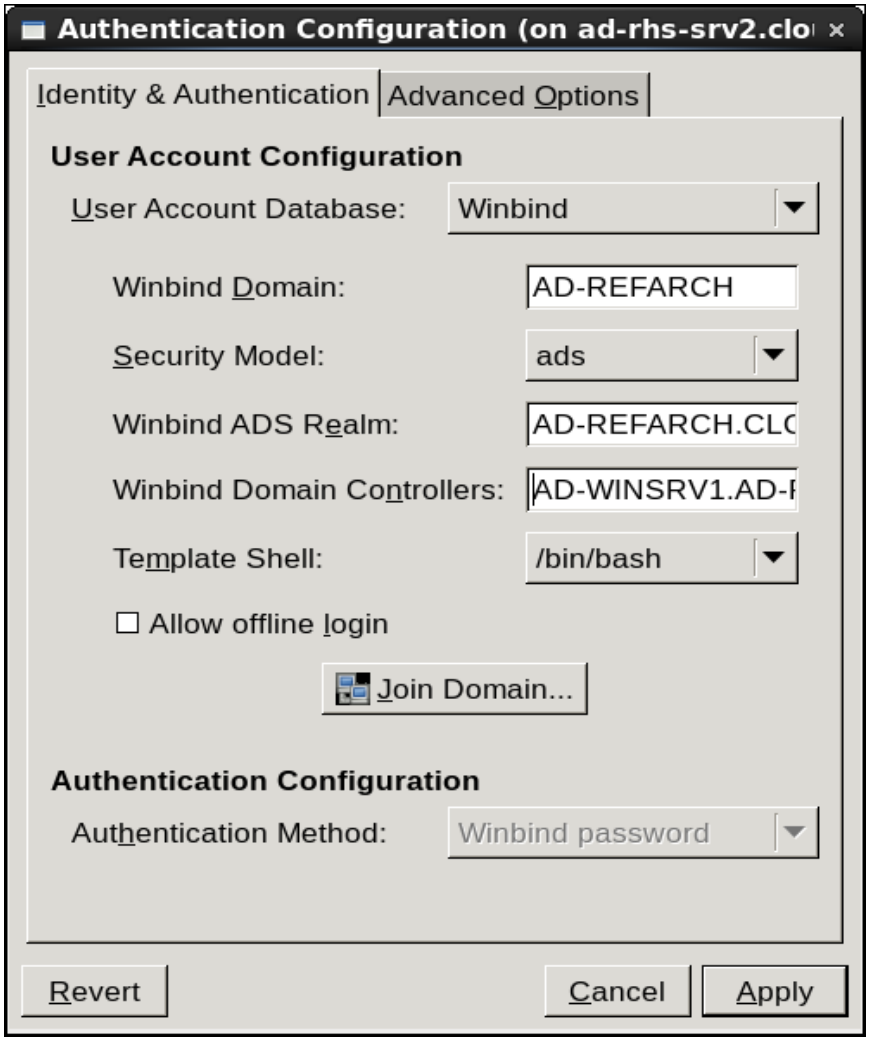


Figure 5.2.5-2: User Account Configuration

Select the Advanced Options tab when done (Optional)

Under **Other Authentication Options**, select **Create home directories on the first login**.

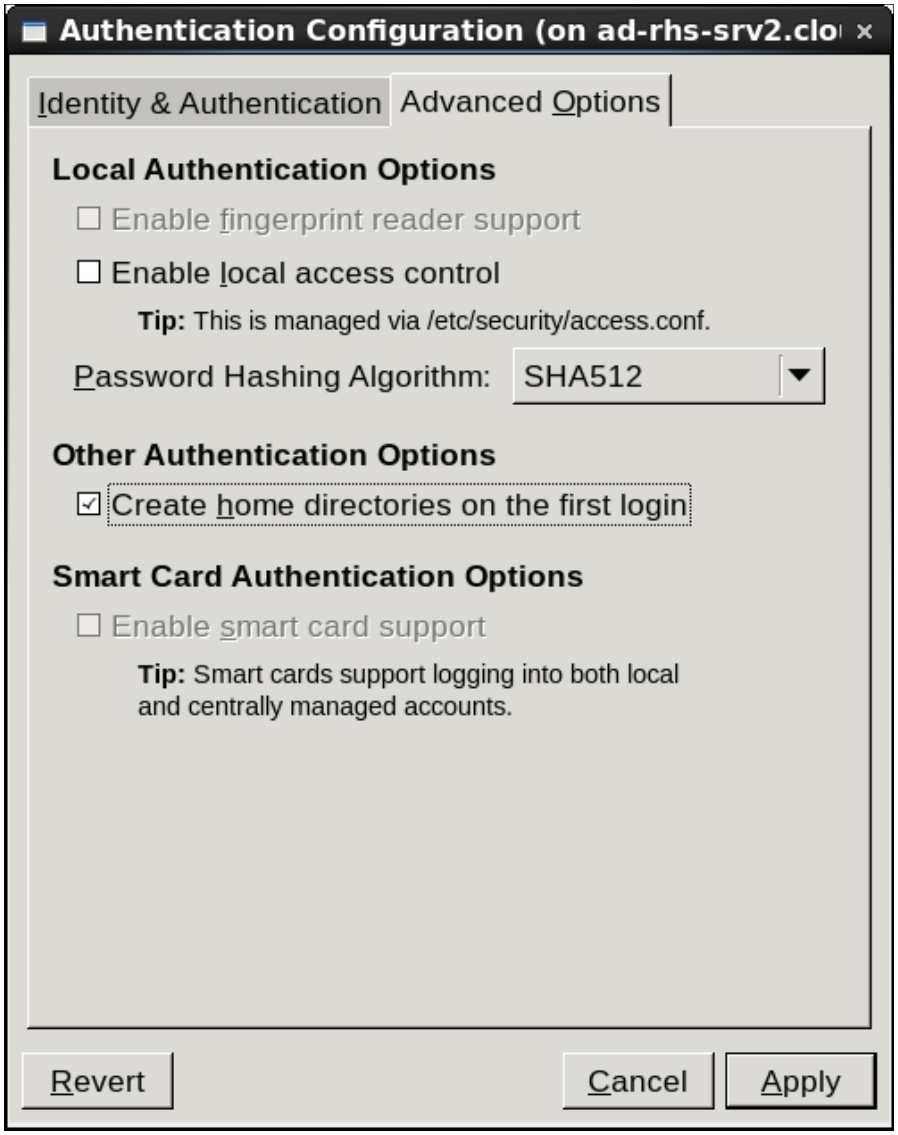


Figure 5.2.5-3: Advanced Options

On the first successful login to Active Directory, the **oddjobd** daemon calls a method to create a new home directory for a user.

Note: This step is not required when clients connect to Red Hat Storage Server just for data access. This is appropriate when the user performs a login to the server and requires a home directory on the server.

Return to the **Identity & Authentication** tab, select **Join Domain**. An alert indicates to save the configuration changes to disk before continuing:

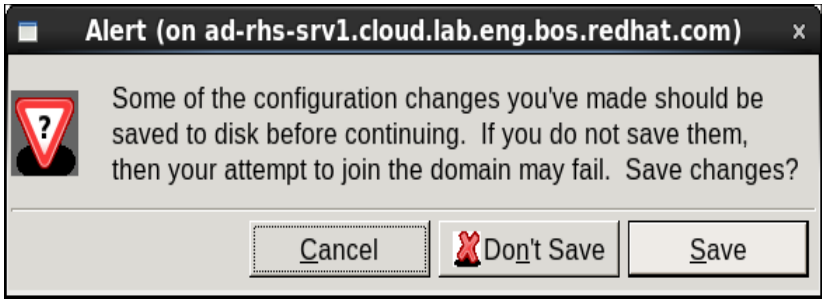


Figure 5.2.5-4: Save Changes

Select **Save**. A new window prompts for the Domain administrator password:

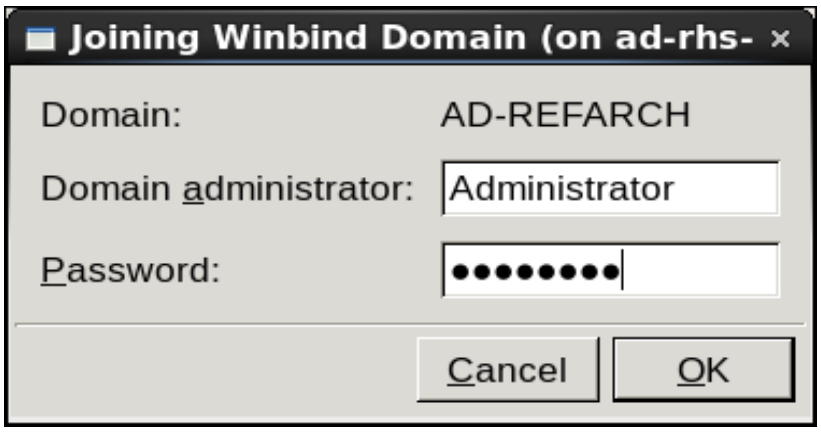


Figure 5.2.5-5: Joining Winbind Domain

Select **OK**. The terminal window displays the status of the domain join:

```
[/usr/bin/net join -w AD-REFARCH -S AD-WINSRV1.AD-
REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM -U Administrator]
Enter Administrator's password:<...>

Using short domain name -- AD-REFARCH
Joined 'AD-RHS-SRV1' to realm 'ad-refarch.cloud.lab.eng.bos.redhat.com'
Not doing automatic DNS update in aclustered setup.
```

Select Apply. The terminal window indicates that Winbind and the oddjobd were started:

```
Starting Winbind services:        [  OK  ]
Starting oddjobd:              [  OK  ]
```

Perform the previous authentication configuration tasks on each of the Red Hat Storage nodes before proceeding to the next section.

5.2.5.2 Verify/Test Active Directory

The join to the Active Directory domain is complete. Verify access by performing each of the following tasks.

Test Connection to AD:

```
# net ads testjoin
Join is OK
```

List members in domain:

```
# wbinfo --domain-users
AD-REFARCH\administrator
AD-REFARCH\guest
AD-REFARCH\krbtgt
AD-REFARCH\test
AD-REFARCH\rhs-user1
AD-REFARCH\rhs-user2
```

List groups in domain:

```
# wbinfo --domain-groups
AD-REFARCH\domain computers
AD-REFARCH\domain controllers
AD-REFARCH\schema admins
AD-REFARCH\enterprise admins
                          ...output abbreviated...
AD-REFARCH\dnsadmins
AD-REFARCH\dnsupdateproxy
AD-REFARCH\rhel-users
```

Note: If either of these fail to return all users or groups in the domain, the idmap UID, GUI upper boundaries in the Samba configuration file need to be increased and the winbind and smb daemons restarted. These tasks are discussed in the next section.

5.2.5.3 Modify Samba Configuration

Next, the Samba configuration file is modified to use the **`idmap_ad`** back-end and several other parameters are configured for convenience. **Table 9: Summary of changes** provides a summary of the configuration file parameter changes:

Samba Configuration File Parameters	
Parameter	**Description**
Netbios name	Set server name AD-RHS-SRV1
idmap uid = 10000-19999	Set user id range for default backend (tdb)
idmap gid = 10000-19999	Set group id range for default backend (tdb)
idmap config AD-REFARCH:backend = ad	Configure winbind to use idmap_ad backend
idmap config AD-REFARCH:default = yes	Configure AD-REFARCH as default domain
idmap config AD-REFARCH:range = 10000000-19999999	Set range for idmap_ad backend
idmap config AD-REFARCH: schema_mode = rfc2307	Enable support for rfc2307 UNIX attributes
winbind nss_info = rfc2307	Obtain user home directory and shell from AD
winbind enum users = no	Disable enumeration of users
winbind enum groups = no	Disable enumeration of groups
winbind separator = +	Change default separator from '\' to '+'
winbind use default domain = yes	Remove need to specify domain in commands
winbind nested groups = yes	Enable nesting of groups in Active Directory

Table 9: Summary of changes

Make a safety copy of the Samba configuration file:

```
# cp -p /etc/samba/smb.conf /etc/samba/smb.conf.back
```

Edit and save the Samba configuration file as follows – changes are highlighted in bold:

```
[global]
   workgroup = AD-REFARCH
   netbios name = AD-RHS-SRV1
   password server = AD-WINSRV1.AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
   realm = AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
   security = ads
   idmap uid = 10000-19999
   idmap gid = 10000-19999
   idmap config AD-REFARCH:backend = ad
   idmap config AD-REFARCH:default = yes
   idmap config AD-REFARCH:range = 10000000-19999999
   idmap config AD-REFARCH:schema_mode = rfc2307
   winbind nss info = rfc2307
   winbind enum users = no
   winbind enum groups = no
   winbind separator = +
   winbind use default domain = yes
   winbind nested groups = yes
```

These changes have to be performed on all Red Hat Storage nodes.

Note: To ensure all the CTDB nodes (Red Hat Storage nodes in this case) can join the AD domain simultaneously, the **netbios name** setting must be the first server name (*ad-rhs-srv1*) on all the nodes. This server acts as the master server from a CTDB point of view, handles the kerberos tickets and shares them with its peers. Hence the **netbios name** should point to *ad-rhs-srv1* in the *smb.conf* file on all the nodes. Failure to do so limits only one of the CTDB nodes to join AD at any point in time.

Test the new configuration file:

```
# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[data1]"
Loaded services file OK.
'winbind separator = +' might cause problems with group membership.
Server role: ROLE_DOMAIN_MEMBER
Press enter to see a dump of your service definitions

[global]
   workgroup = AD-REFARCH
   realm = AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
   server string = Samba Server Version %v
   security = ADS
   password server = AD-WINSRV1.AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
   log file = /var/log/samba/log.%m
   max log size = 50
   max protocol = SMB2
   clustering = Yes
   load printers = No
   disable spoolss = Yes
   show add printer wizard = No
   stat cache = No
   kernel oplocks = No
   template shell = /bin/bash
```

```
   winbind separator = +
   winbind use default domain = Yes
   winbind nss info = rfc2307
   idmap config AD-REFARCH:schema_mode = rfc2307
   idmap config AD-REFARCH:range = 10000000-19999999
   idmap config AD-REFARCH:default = yes
   idmap config AD-REFARCH:backend = ad
   idmap config * : range = 10000-19999
   idmap config * : backend = tdb2
   printing = bsd
   print command = lpr -r -P'%p' %s
   lpq command = lpq -P'%p'
   lprm command = lprm -P'%p' %j
   map archive = No
   map readonly = no
   store dos attributes = Yes

[gluster-dfilevol01]
   comment = For samba share of volume dfilevol01
   path = /
   read only = No
   guest ok = Yes
   vfs objects = glusterfs
   glusterfs:loglevel = 7
   glusterfs:logfile = /var/log/samba/glusterfs-dfilevol01.log
   glusterfs:volume = dfilevol01

[gluster-ctdbmeta]
   comment = For samba share of volume ctdbmeta
   path = /
   read only = No
   guest ok = Yes
   vfs objects = glusterfs
   glusterfs:loglevel = 7
   glusterfs:logfile = /var/log/samba/glusterfs-ctdbmeta.log
   glusterfs:volume = ctdbmeta
```

Backup and clear out the existing Samba cache files - requires services to be stopped:

```
# service smb stop
Shutting down SMB services:                [  OK  ]

# service winbind stop
Shutting down Winbind services:            [  OK  ]

# tar -cvf /var/tmp/samba-cache-backup.tar /var/lib/samba
tar: Removing leading `/' from member names
/var/lib/samba/
/var/lib/samba/smb_krb5/
/var/lib/samba/smb_krb5/krb5.conf.AD-REFARCH

                          ...output abbreviated...

/var/lib/samba/registry.tdb
/var/lib/samba/perfmon/
/var/lib/samba/winbindd_idmap.tdb
```

```
# ls -la /var/tmp/samba-cache-backup.tar
-rw-r--r--. 1 root root 512000 May 20 17:06 /var/tmp/samba-cache-backup.tar
# rm -f /var/lib/samba/*
```

Verify no Kerberos tickets are in use:

```
# kdestroy
kdestroy: No credentials cache found while destroying cache
# klist
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

Join the Active Directory domain:

```
# net join -S ad-winsrv1 -U administrator
Enter administrator's password:
Using short domain name -- AD-REFARCH
Joined 'AD-RHS-SRV1' to realm 'ad-refarch.cloud.lab.eng.bos.redhat.com'
Not doing automatic DNS update in aclustered setup.
```

Test connection to the Active Directory domain:

```
# net ads testjoin
Join is OK
[root@ad-rhs-srv2 ~]# net ads info
LDAP server: 10.16.136.53
LDAP server name: ad-winsrv1.ad-refarch.cloud.lab.eng.bos.redhat.com
Realm: AD-REFARCH.CLOUD.LAB.ENG.BOS.REDHAT.COM
Bind Path: dc=AD-REFARCH,dc=CLOUD,dc=LAB,dc=ENG,dc=BOS,dc=REDHAT,dc=COM
LDAP port: 389
Server time: Thu, 16 May 2013 18:01:48 EDT
KDC server: 10.16.136.53
Server time offset: 0
```

Start Winbind and Samba to activate the new configuration changes:

```
# service winbind start
Starting Winbind services:                 [  OK  ]
# service winbind status
winbindd (pid  24416) is running...
# ps -aef | grep winbind
root     24416     1  0 17:12 ?        00:00:00 winbindd
root     24421 24416  0 17:12 ?        00:00:00 winbindd
root     24484 24416  0 17:12 ?        00:00:00 winbindd
root     24487 24416  0 17:12 ?        00:00:00 winbindd
root     24489 24416  0 17:12 ?        00:00:00 winbindd

# service smb start
Starting SMB services:                     [  OK  ]
# service smb status
smbd (pid  24482) is running...
# ps -aef | grep smbd
root     24482     1  0 17:12 ?        00:00:00 smbd -D
root     24495 24482  0 17:12 ?        00:00:00 smbd -D
```

List members in domain:

```
# wbinfo --domain-users
administrator
guest
krbtgt
test
rhs-user1
rhs-user2
```

List groups in domain:

```
# wbinfo --domain-groups
domain computers
domain controllers
schema admins
enterprise admins
                          ...output abbreviated...
dnsadmins
dnsupdateproxy
rhel-users
```

5.2.5.4 Verification of Services

Verify the services provided by performing the tasks outlined in the following sections:

1. Login Access - Verify access from another Red Hat Enterprise Linux 6 system, using a different Active Directory user account:

```
$ hostname
ad-rhel1.cloud.lab.eng.bos.redhat.com
$ ssh rhs-user1@rhs-srv
Warning: Permanently added 'ad-rhs-srv1,10.16.141.1' (RSA) to the list of
known hosts.
rhs-user1@rhs-srv's password:
$ id
uid=10000011(rhs-user1) gid=10000011(rhel-users) groups=10000011(rhel-
users)
$ hostname
ad-rhs-srv1.cloud.lab.eng.bos.redhat.com
$ pwd
/home/ad-refarch/rhs-user1
$ echo $SHELL
/bin/bash
$ exit
logout
Connection to rhs-srv closed.
```

Reconnect to the same user and server once again:

```
# ssh rhs-user1@rhs-srv
Warning: Permanently added 'rhs-srv,10.16.141.11' (RSA) to the list of
known hosts.
rhs-user1@rhs-srv's password:
Last login: Wed May 29 11:48:18 2013 from ad-rhs-srv1
$ hostname
ad-rhs-srv2.cloud.lab.eng.bos.redhat.com
$ id
uid=10000011(rhs-user1) gid=10000011(rhel-users) groups=10000011(rhel-
users)
$ pwd
/home/ad-refarch/rhs-user1
```

Note: This time the DNS pointed *rhs-srv* to *ad-rhs-srv2.* However there is no variation in behavior.

6 Accessing Red Hat Storage using Active Directory

6.1 Share Directory

In this reference architecture, directories created under Red Hat Storage Volumes are accessed by clients with users and groups managed by Active Directory server. Multiple directories may be created under a Red Hat Storage Volume to suit to different users, groups or permissions.

6.1.1 Samba Mounts

The user, group and access privileges for the share directories are set in the Red Hat Storage nodes. For these privileges to be set, samba mount for the volumes (For Ex: */mnt/samba/Gluster-vol*) is a requirement. In the current RHS2.1 version, the samba mount is not automated with the start and stop of a Red Hat Storage Volume. In this reference architecture, the samba mounts were automated for volume start and stop by editing the hook scripts:

/var/lib/glusterd/hooks/1/start/post/S30samba-start.sh

/var/lib/glusterd/hooks/1/stop/pre/S30samba-stop.sh

Details related to this change and required modifications are depicted in

Appendix F Modification of hook scripts for Red Hat Storage Volumes

> **Note**: In the previous version RHS2.0, starting a Red Hat Volume automatically appends volume related configurations in /etc/samba/smb.conf, mount point configurations in /etc/fstab and mounts the samba mount point. Also by stopping a volume, all these changes are undone. The modification of the hook scripts restore this functionality in RHS2.1. There is an open Request for Feature Enhancement RFE#1012687 to enable this in future RHS releases.

6.1.2 Creating share directory

Create a sub-directory ‘rhsdata01’ under the samba mount */mnt/samba/dfilevol01*. This is used as a shared directory to the clients. The rhsdata01 directory described below, belongs to the “rhel-users “ AD group.

The following commands must be performed only on the first node and the results are reflected on all the nodes.

```
# mkdir /mnt/samba/dfilevol01/rhsdata01
# chgrp "AD-REFARCH+rhel-users" /mnt/samba/dfilevol01/rhsdata01
# chmod 770 /mnt/samba/dfilevol01/rhsdata01
# ll -d /mnt/samba/dfilevol01/rhsdata01/
drwxr-xr-x 2 770 rhel-users 12 Sep 19 10:10 /mnt/samba/dfilevol01/rhsdata01/
```

> **Note**: The permission and group settings are based on business requirements.

6.1.3 ACL Settings

ACL imparts more granularity than standard chmod permissions in terms of providing better access to users and groups. The ACL can be set for a file or directory and can be set according to user/business requirements.

Change directory to the brick location.

```
# cd /mnt/samba/dfilevol01/
# setfacl -d -m g:"AD-REFARCH+rhel-users":rwx rhsdata01
# setfacl -d -m g::--- rhsdata01
```

Verify the ACL settings

```
# getfacl /mnt/samba/dfilevol01/rhsdata01
getfacl: Removing leading '/' from absolute path names
# file: mnt/samba/test/rhsdata01
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::---
default:group:rhel-users:rwx
default:mask::rwx
default:other::r-x
```

Note: To get ACL to work, both the parent filesystem (/rhs/storage1) and the samba mount (/mnt/samba/dfilevol01) must have '**acl**' enabled.
For more details on ACL please refer to *https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Storage_Administration_Guide/s1-acls-setting.html*

6.2 Accessing from Windows Clients

6.3 Accessing from a Windows 2008 R2 Client

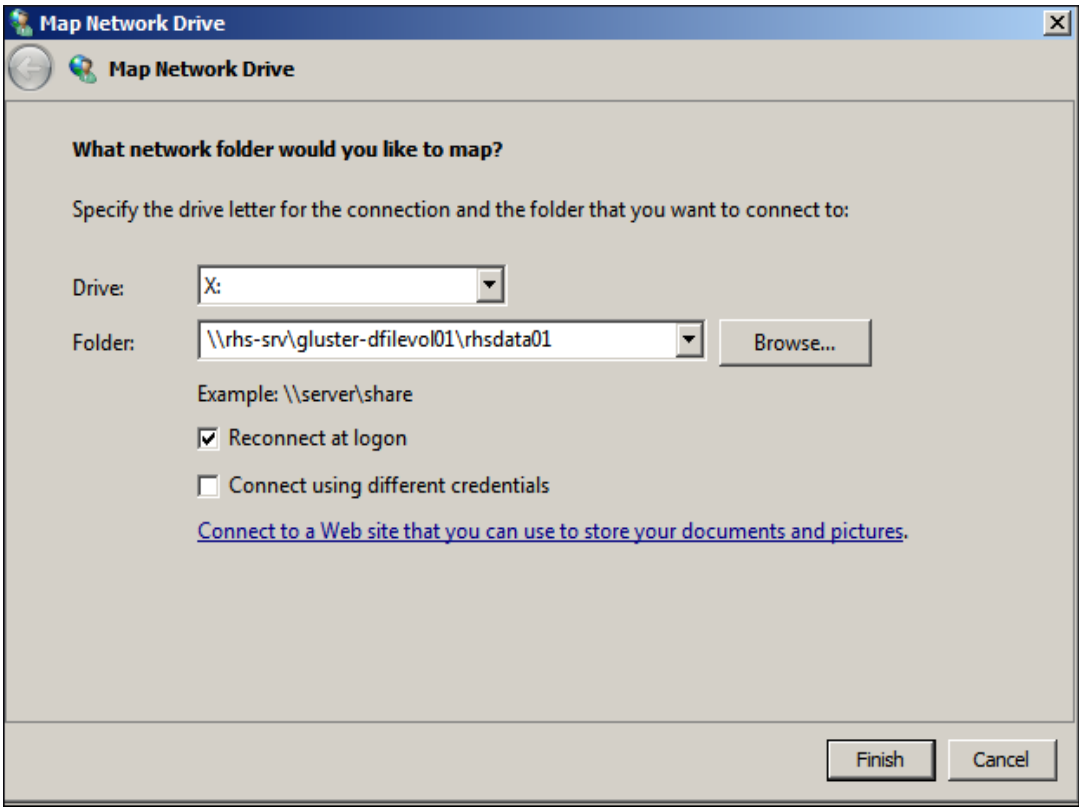


Figure 6.3-1: Mapping the Drive

Map a drive on Windows client *AD-WIN2K8* running Windows 2008 R2 pointing to the share directory created in **Section 6.1.2Creating share directory**.Provide user credentials that belongs to the AD domain.

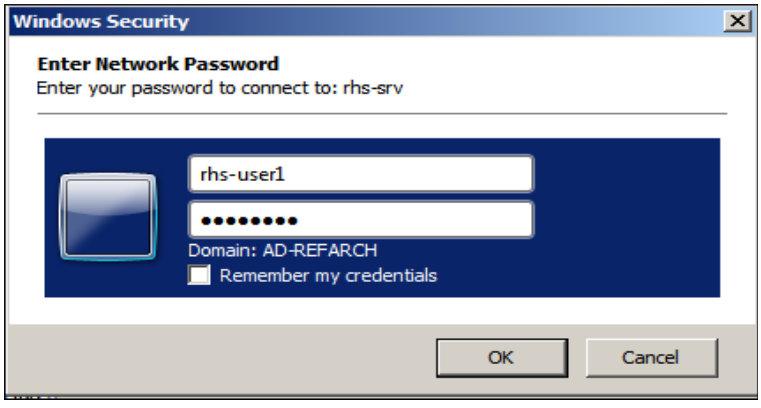


Figure 6.3-2: User Credentials

Upon successful mapping of the drive, the details can be viewed in the *Computer* window.

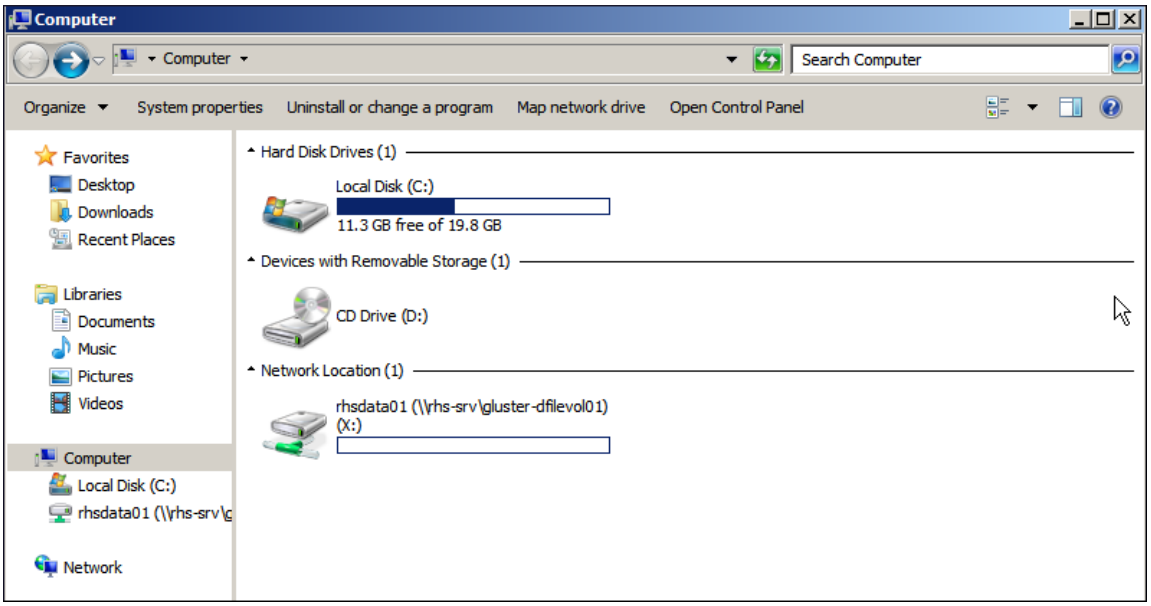


Figure 6.3-3: New Mapped Drive

Double click on this mapped drive and create a new test file testfile-ad-win2k8

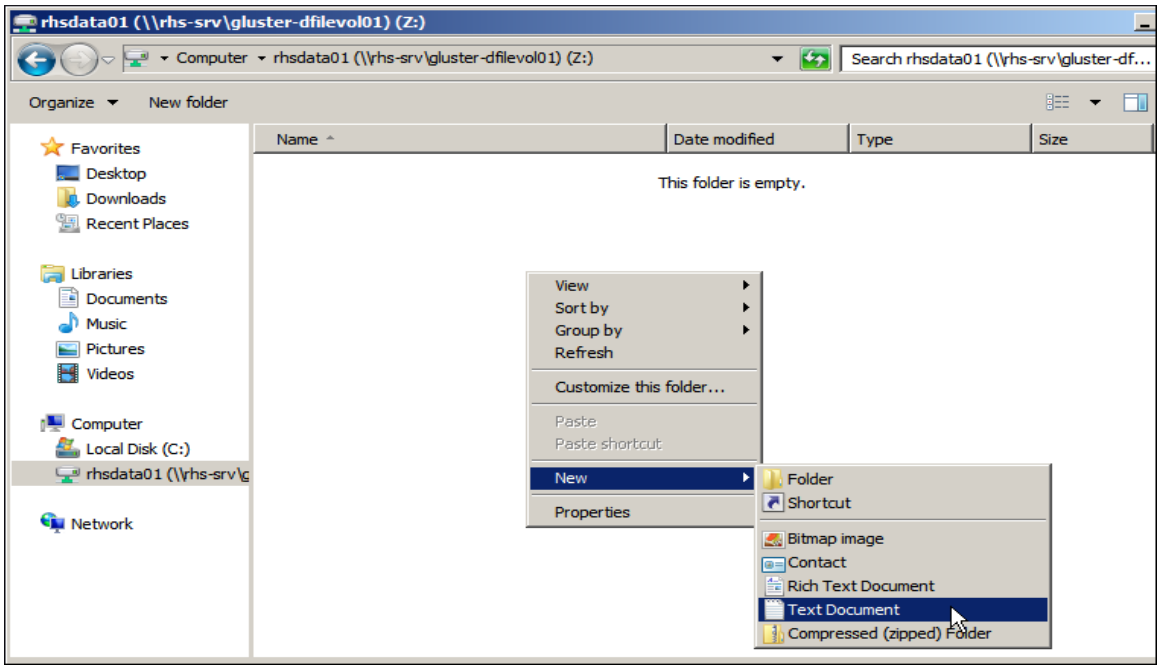


Figure 6.3-4: Creating a New File - 1

Add the following content to the file from the client.

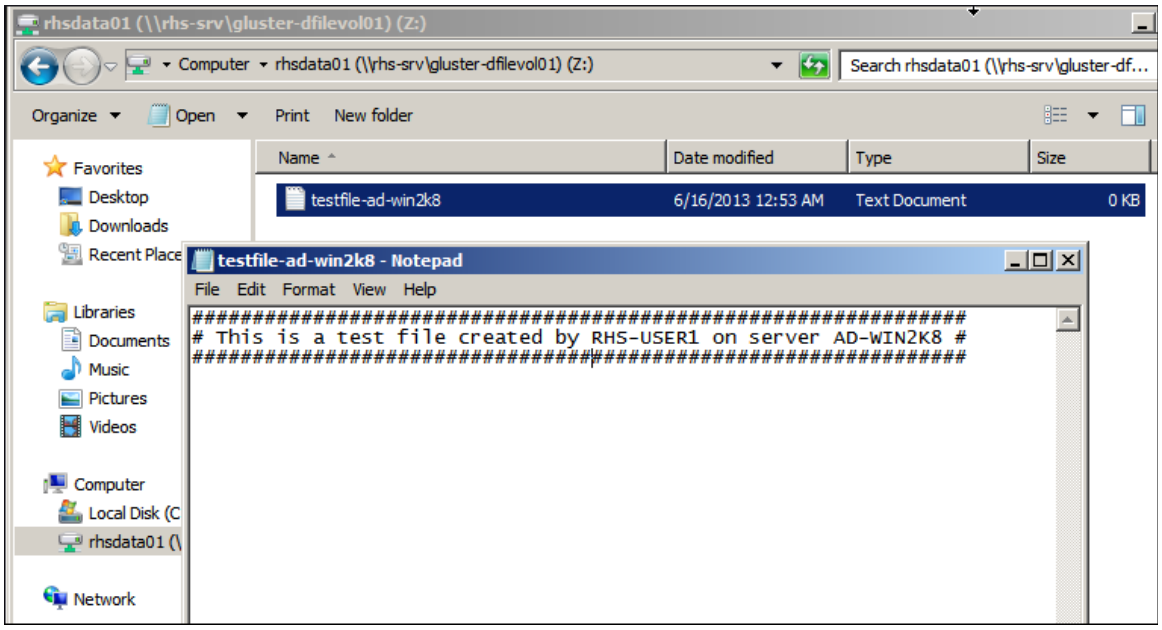


Figure 6.3-5: Creating a New File - 2

Access the file */mnt/samba/dfilevol01/rhsdata01/testfile-ad-win2k8* from any of the Red Hat Storage nodes and update the file. View the updated information from the client window.

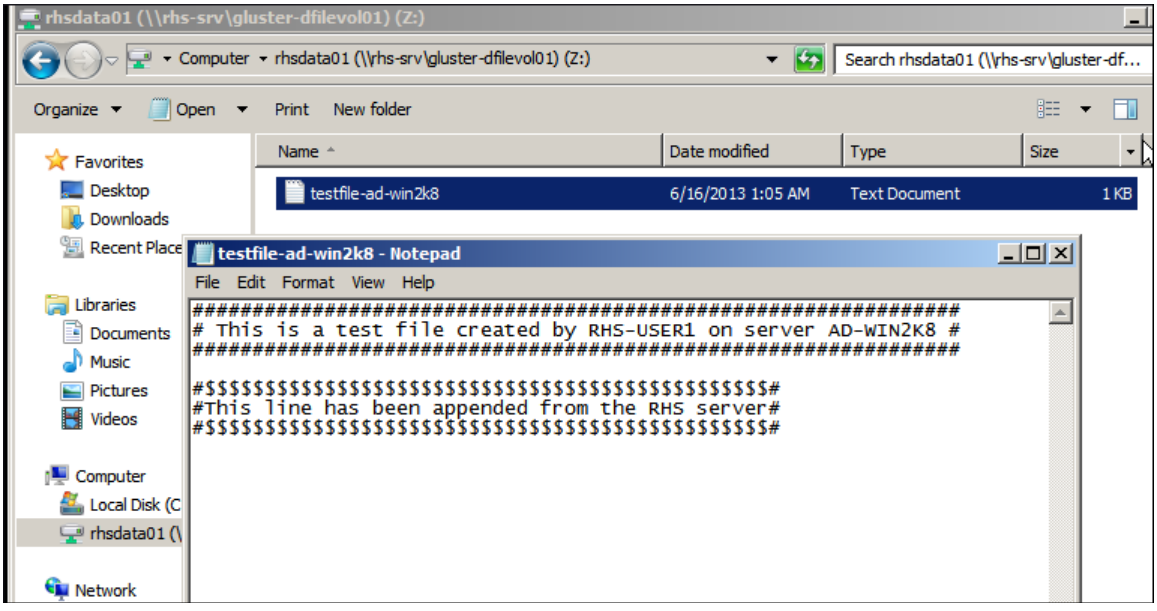
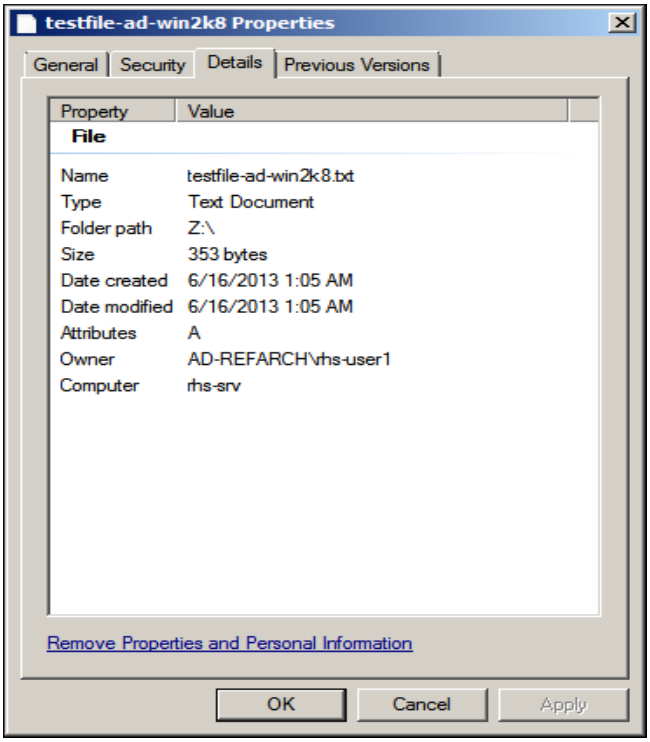


Figure 6.3-6: File after editing from RHS Server



Review the file properties on the client:

Figure 6.3-7: File Properties

ACL command output for the file from Windows Client:

```
C:\>cacls Z:\testfile-ad-win2k8.txt
Z:\testfile-ad-win2k8.txt AD-REFARCH\rhs-user1:(special access:)
                                         READ_CONTROL
                                       WRITE_DAC
                                       WRITE_OWNER
                                       SYNCHRONIZE
                                       FILE_GENERIC_READ
                                       FILE_GENERIC_WRITE
                                       FILE_GENERIC_EXECUTE
                                       FILE_READ_DATA
                                       FILE_WRITE_DATA
                                       FILE_APPEND_DATA
                                       FILE_READ_EA
                                       FILE_WRITE_EA
                                       FILE_EXECUTE
                                       FILE_DELETE_CHILD
                                       FILE_READ_ATTRIBUTES
                                       FILE_WRITE_ATTRIBUTES

                    AD-REFARCH\rhel-users:(special access:)
                                        READ_CONTROL
                                        SYNCHRONIZE
                                        FILE_GENERIC_READ
                                        FILE_READ_DATA
                                        FILE_READ_EA
                                        FILE_READ_ATTRIBUTES
                        Everyone:(special access:)
```

Review the file attributes from the Red Hat Storage node:

```
# getfacl /mnt/samba/dfilevol01/rhsdata01/testfile-ad-win2k8.txt
# file: testfile-ad-win2k8.txt
# owner: rhs-user1
# group: rhel-users
# flags: --t
user::---
group::---
group:domain\040users:rwx
mask::rwx
other::---
```

6.3.1 Accessing from a Windows 7 Client

There is no functional difference between Windows 7 client and Windows 2008 R2, with respect to accessing Red Hat Storage. The following figure displays mapping a drive.

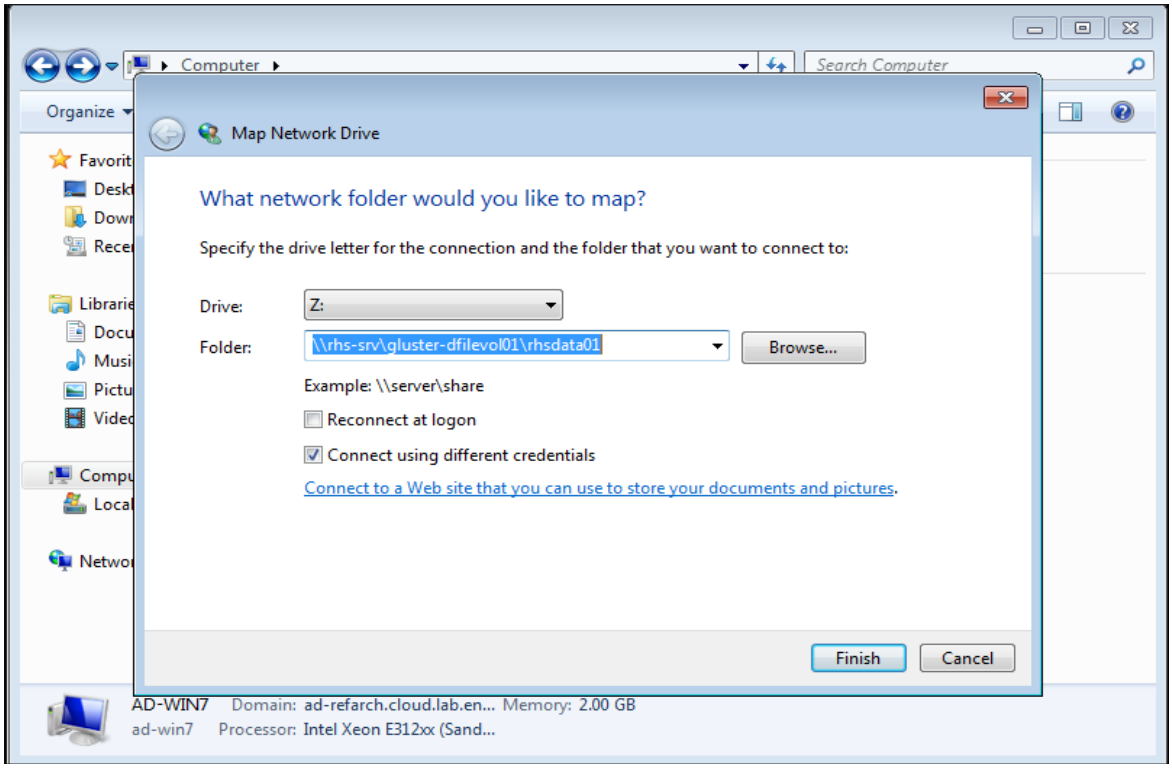


Figure 6.3.1-1: Accessing Red Hat Storage from Windows 7 -1

The following figure displays a file created at Windows client on a Red Hat Storage shared directory.

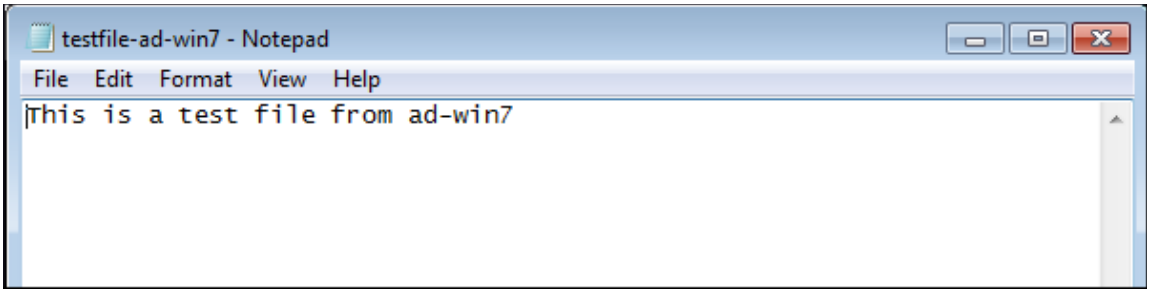


Figure 6.3.1-2: Accessing Red Hat Storage from Windows 7 -2

File properties as seen from a Windows 7 client.

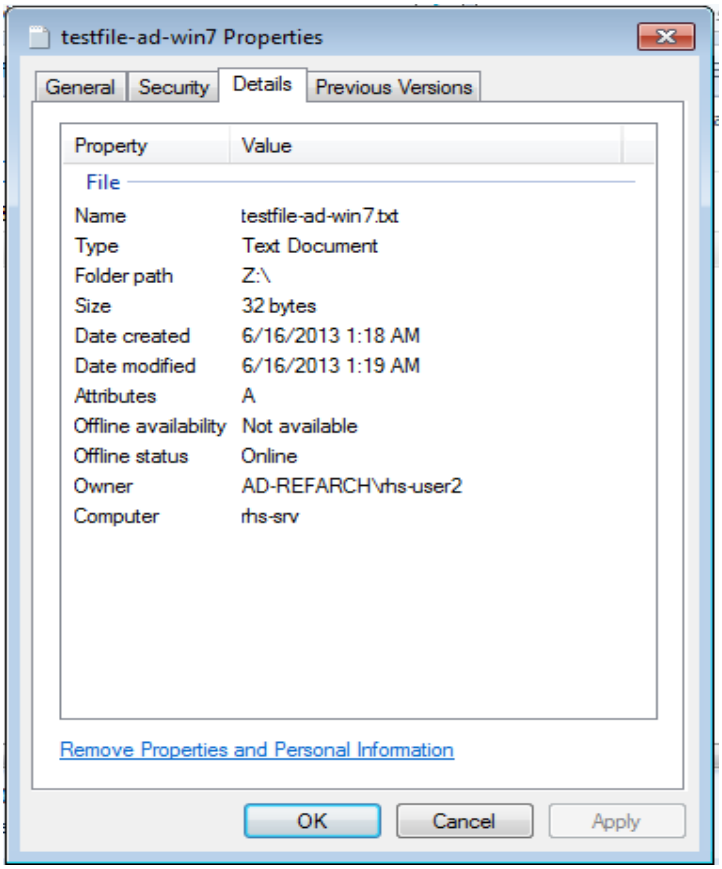


Figure 6.3.1-3: Accessing Red Hat Storage from Windows 7 -3

6.4 Accessing from Red Hat Enterprise Linux Clients

RHEL Clients can access regular files in Red Hat Storage Server in three ways:

- Native Client using FUSE
- NFS v3
- SMB (Server Message Block) using CIFS

This reference architecture describes connectivity using SMB.
The RHEL clients must be added to the Active Directory domain as described in **SectionB.5Add Red Hat Storage Server to AD Domain** .

The following packages are required to be installed on the RHEL client:

```
# yum install samba-winbind
# yum install samba-client
# yum install krb5-workstation
# yum install cifs-utils-4.4-5.el6.x86_64
```

Mount the CIFS filesystem with user parameters set in the credential file */rhs/cred:*

```
# mount -t cifs rhs-srv:/gluster-dfilevol01 -o credential=/rhs/cred
/mnt/rhsdata
```

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/myvg-rootvol               133G  120G  6.9G  95% /
tmpfs                                  16G     0   16G   0% /dev/shm
/dev/sda1                              194M   55M  130M  30% /boot
//rhs-srv/gluster-dfilevol01/          9.1T   98M  9.1T   1% /mnt/rhsdata
```

where **gluster-dfilevol01** is the exported storage volume through SMB from *rhs-srv* using the */etc/samba/smb.conf* file with the following entry:

```
[gluster-dfilevol01]
comment=For samba export of volume dfilevol01
path=/mnt/dfilevol01
read only=no
guest ok=yes
```

The user credentials are stored in the file */rhs/cred.* This credential file must be created with the below mentioned contents and secured with the right permissions:

```
username=rhs-user1
password=<password>                          (password for user rhs-user1)
domain=ad-refarch
```

Note: The username and password are managed by the AD server with ‘ad-refarch’ as the domain.

This completes the process of integrating Red Hat Storage nodes into an Active Directory domain.

7 Conclusion

This reference architecture details the components, considerations and configurations available for selecting, deploying, and integrating Red Hat Storage Servers into Windows Active Directory domains. Basic concepts are introduced, deployment and integration tasks outlined, and best practices and guidelines provided.

These configurations can be deployed as presented here, or customized to meet the specific requirements of system administrators wanting to integrate Red Hat Storage Servers and Red Hat Enterprise Linux clients into their existing Microsoft Windows Active Directory domain environments.

Appendix A: Contributors

Contributor	Title	Contribution
Veda Shankar	Senior Principal Product Marketing Manager	RHS Deployment Assistance
Peter Portante	Principal Software Engineer	RHS Performance
Ben England	Principal Software Engineer	RHS Performance
Sayandeb Saha	Manager, Product Management	Review
Tushar Katarki	Principal Product Manager-Technical	Review

Appendix B: Active Directory Domain Services – Configuration

This summary is provide as a guide to the installation and configuration of Active Directory Domain Services on Windows Server 2008 R2.

B.1 Prerequisites

The following are required before Active Directory can be configured on a Windows Server 2008 R2 server:

- Windows activation
- Administrator account access
- Properly configured NIC with static IP
- NTFS partition with a minimum 250mb of free space for Active Directory
- Functional DNS server (can be installed on the AD server itself or point to an existing DNS server)
- Dedicated domain name to use

B.2 Installation Summary

Refer to the following Microsoft TechNet article for the most current and comprehensive details: http://technet.microsoft.com/en-us/library/dd378801%28v=ws.10%29.aspx

An Active Directory installation involves the following series of steps on a Windows Server 2008 R2 server:

1. Install Active Directory Domain Services Role
2. Configure Active Directory Domain Services
3. Configure Windows Time Service
4. Create DNS Forward and Reverse Lookup Zones
5. Restart DNS Service
6. Verify Active Directory Domain Services
7. Create User Accounts
8. Verify Client Access to Active Directory Domain
9. Add Red Hat Enterprise Linux 6 Server DNS A Record (optional)

Details on each of these steps are provided in the next section.

B.3 Installation Details

1. Install Active Directory Domain Services Role
   - Open *Server Manager:* **Start -> Administrative Tools -> Server Manager**
   - Select **Roles -> Add Roles.** The *Add Roles* Wizard opens. Select **Next** to continue.

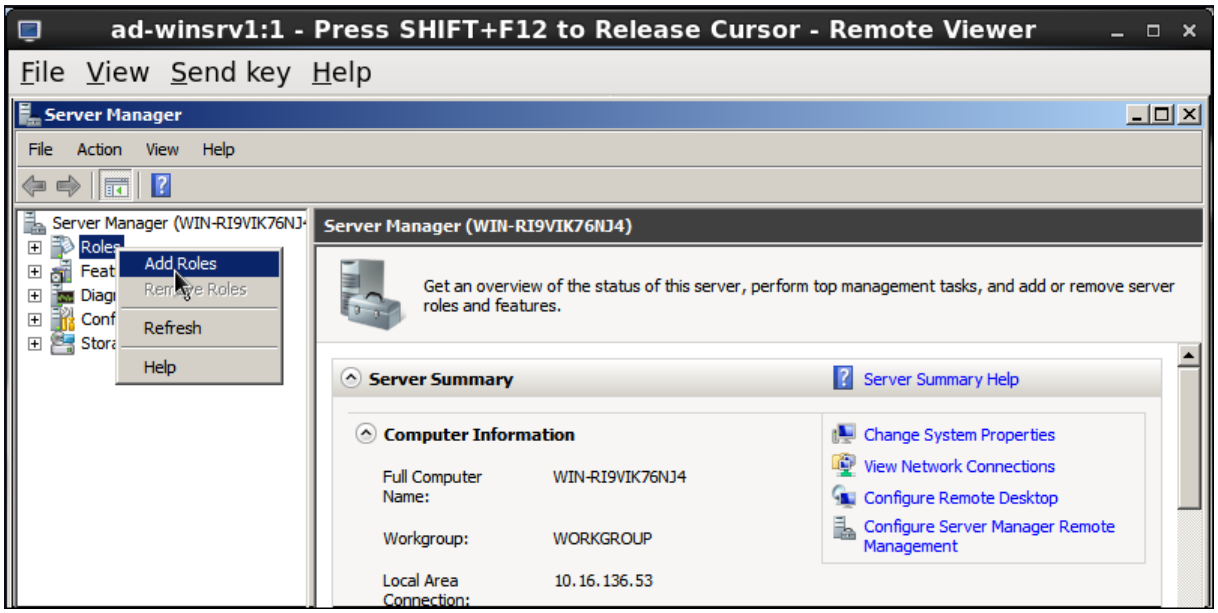


Figure B.3-1: Active Directory - Add Roles

- Under *Roles* select **Active Directory Domain Services**

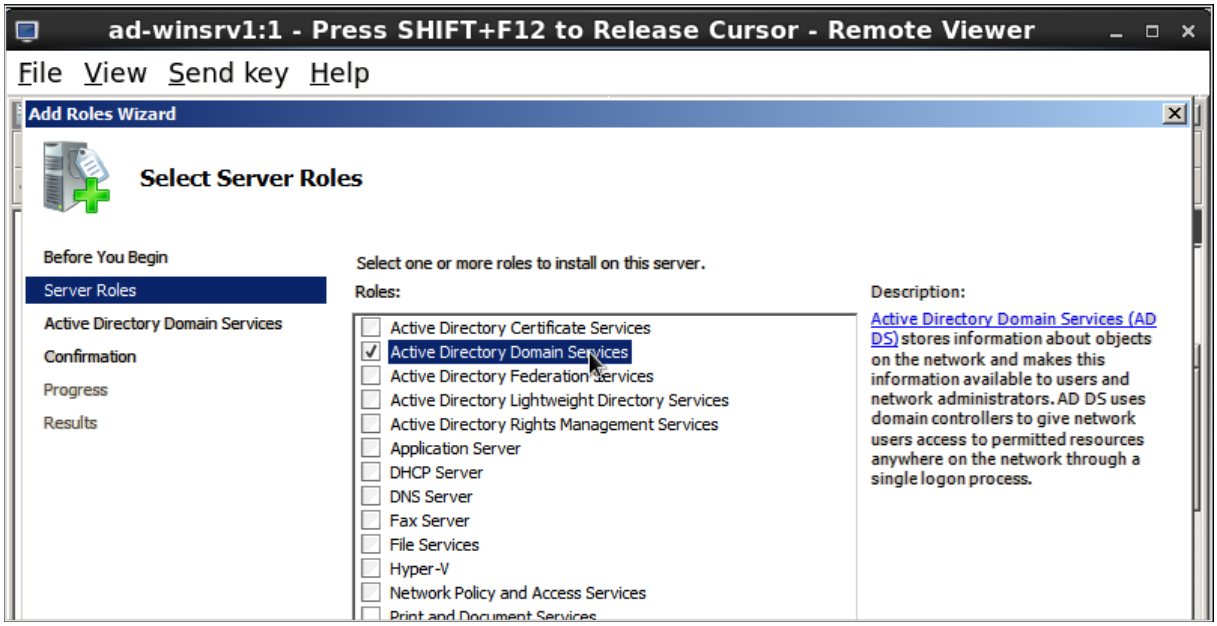


Figure B.3-2: Active Directory - Domain Services 1

- Select **Next** to proceed with the install

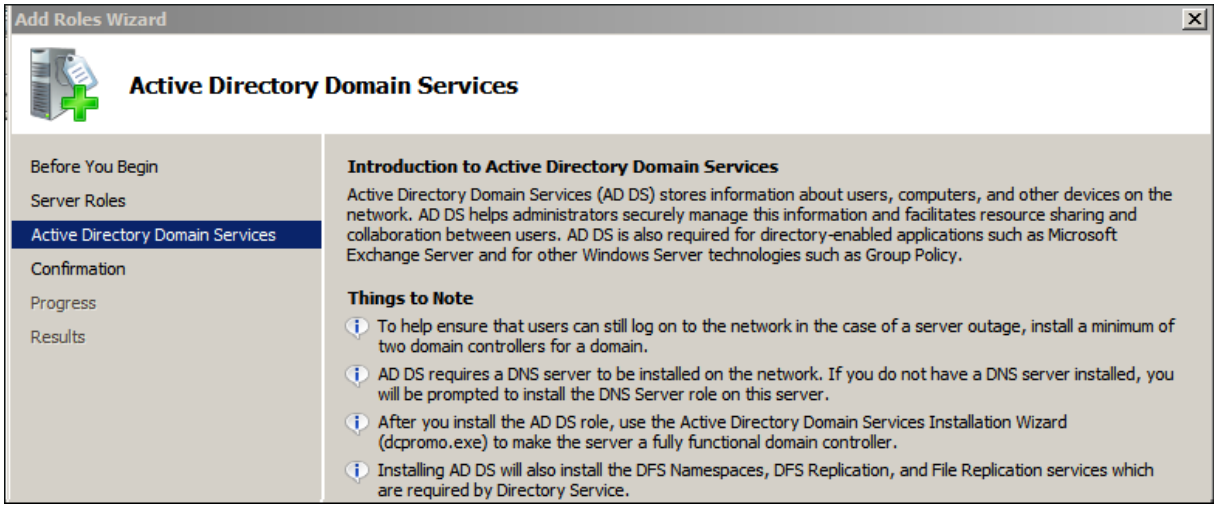


Figure B.3-3: Active Directory Domain Services 2

Note: If .NET Framework 3.5.1 is not installed, a prompt appears asking whether or not to install it. Select **Install** to continue.

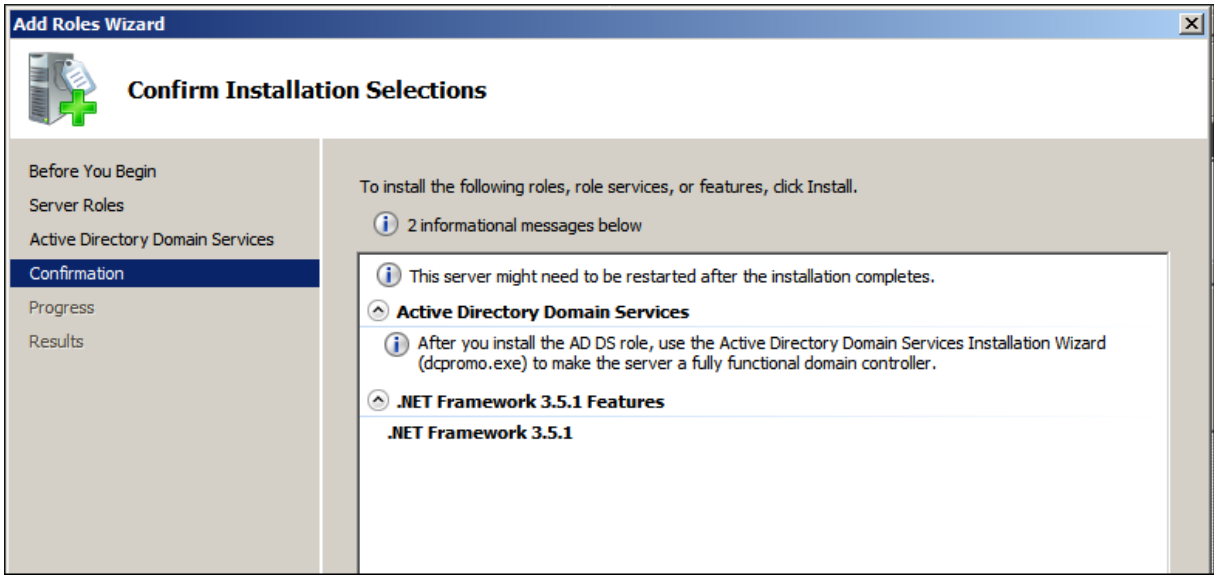


Figure B.3-4: Active Directory Domain Services 3

- Select **close** after confirming the Active Directory Domain Services (and if applicable .Net Framework 3.5.1) Installation Results.

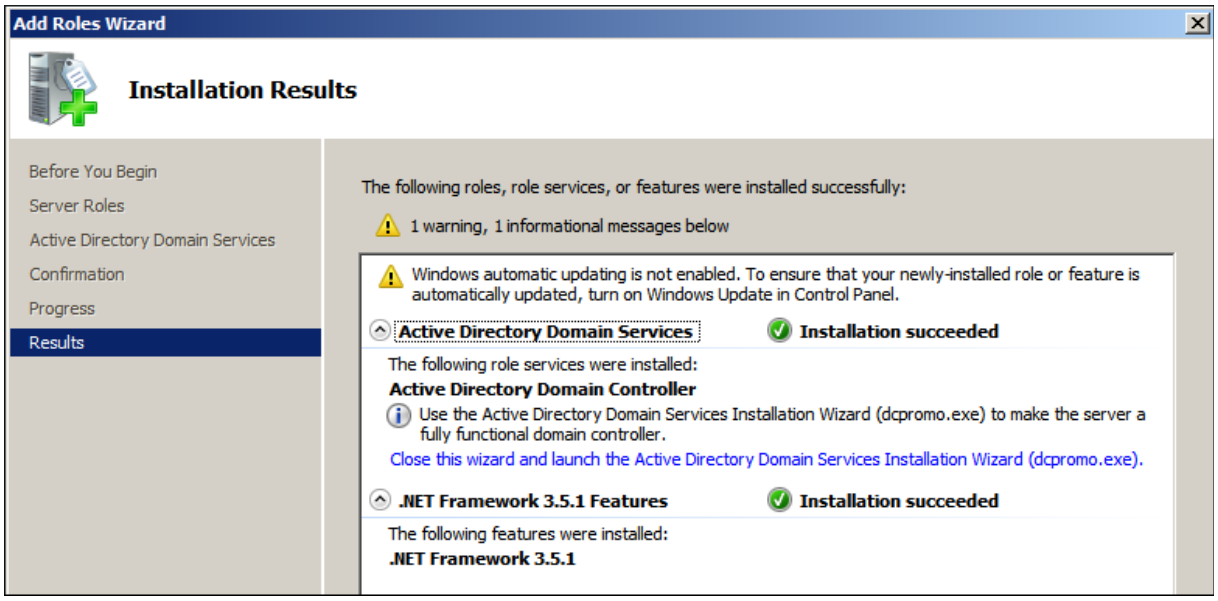


Figure B.3-5: Active Directory Domain Services 4

2. Configure Active Directory Domain Services
   - Under *Roles*, select the **Active Directory Domain Services** link
   - At the top of the Summary section select the **Run the Active Directory Domain Services Installation Wizard** (dcpromo.exe) link

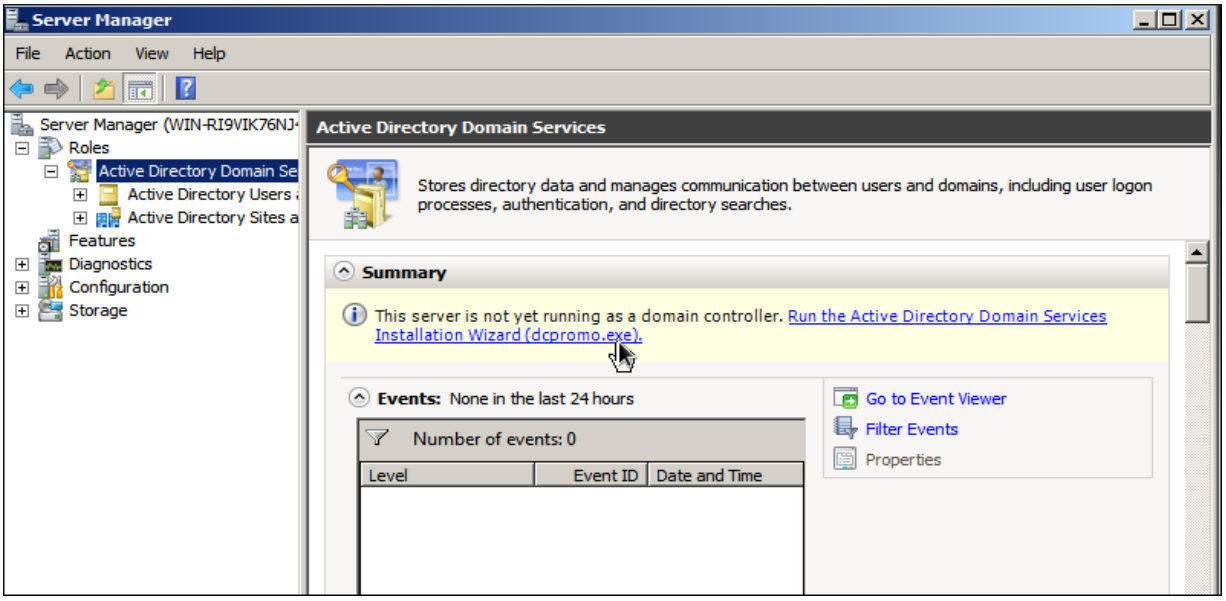


Figure B.3-6: Active Directory Domain Services Configuration

- Select **Next** to display *Active Directory Domain Services Installation Wizard*
- In the *Choose a Deployment Configuration window* select **Create a new domain in a new forest**, then select **Next**

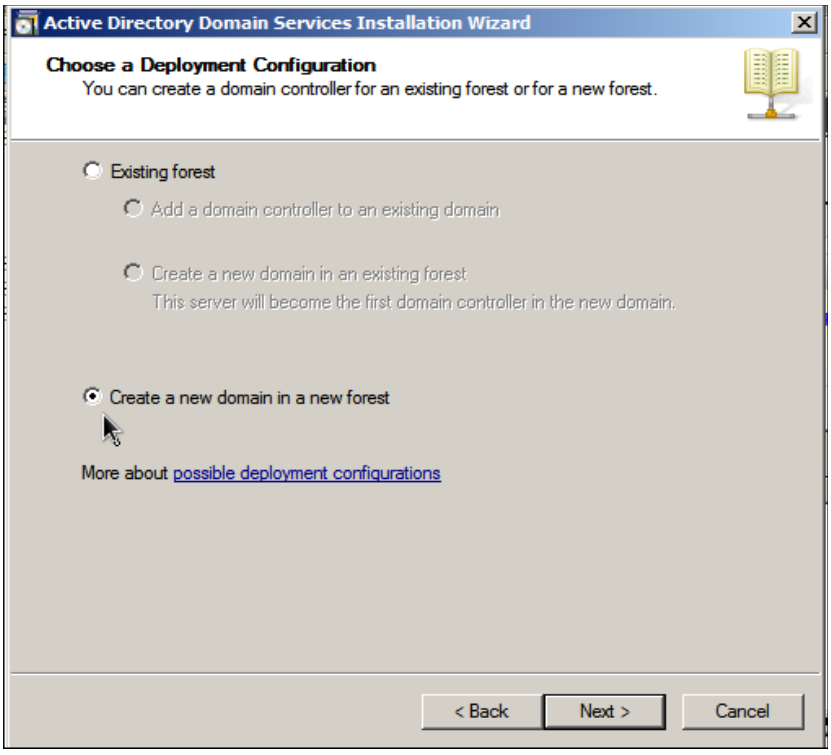


Figure B.3-7: Active Directory New Domain

- Enter the *Fully Qualified Domain Name (FQDN)* of the new forest domain

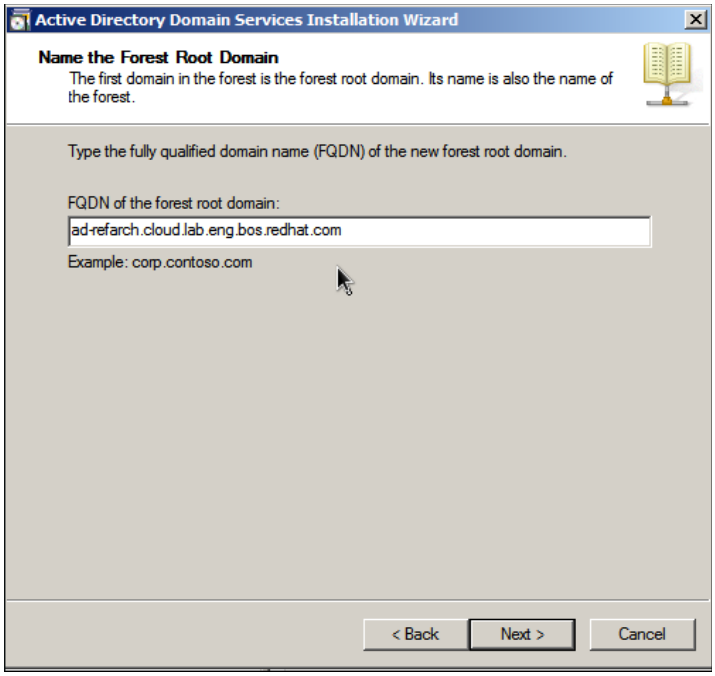


Figure B.3-8: Active Directory FQDN

Note: Single label domain names - e.g. **ad-refarch** must not be used, instead a fully qualified domain name must be used– e.g. **ad-refarch.cloud.lab.eng.bos.redhat.com**

- Select **Next** to continue after the wizard has verified the domain name is not already in use on the local network
- Select the appropriate *Forest functional level - Windows Server 2008 R2*.
- Select **Next** to continue

Note: If this is a forest in an existing domain then select the appropriate minimum server level appropriate to the environment.

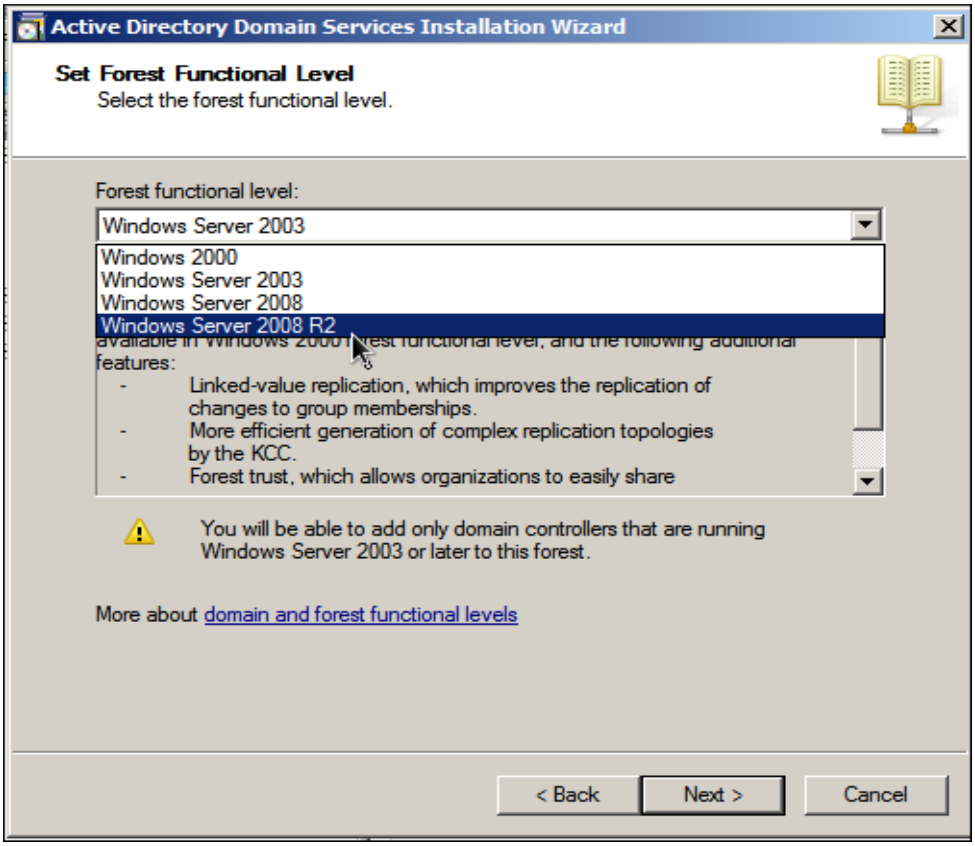


Figure B.3-9: Active Directory Functional Level

- In the Additional Domain Controller Options window, ensure **DNS server** is selected then select **Next**

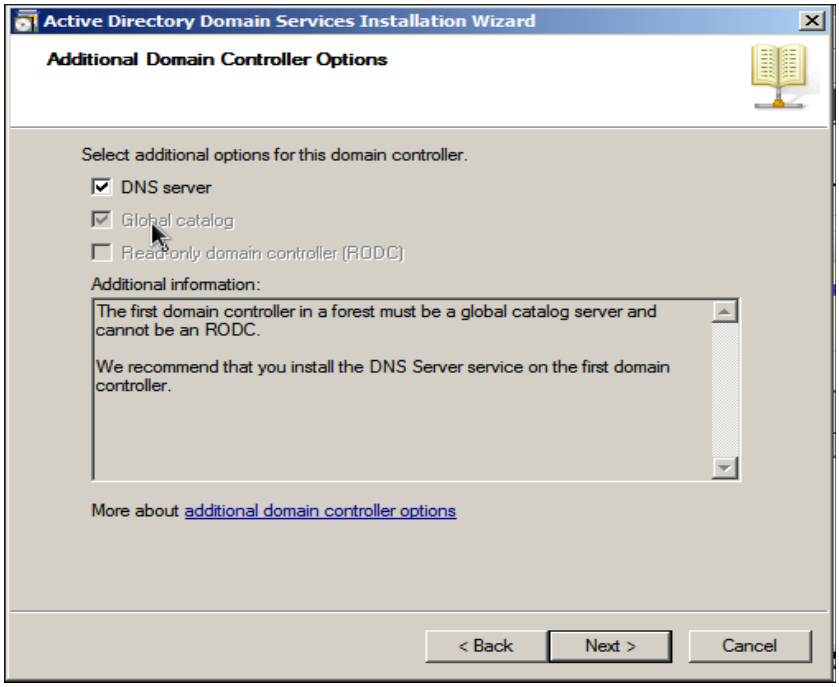


Figure B.3-10: Active Directory DNS Server 1

- If a static IP address was not previously configured, then the *Static IP Assignment* window warns "*This computer had dynamically assigned IP address(es)*" if one or more network interfaces is set to a dynamic IP.
- Depending on your configuration select either of the following options below:

"Yes, the computer will use an IP address automatically assigned by a DHCP server (not recommended)"

...or...

"No, I will assign static IP addresses to all physical network adapters"

Note: For production servers it is highly recommended that static IP addresses be used.

- The Active Directory Domain Services Installation Wizard warns that no DNS has been configured yet. Select **Yes** to continue.

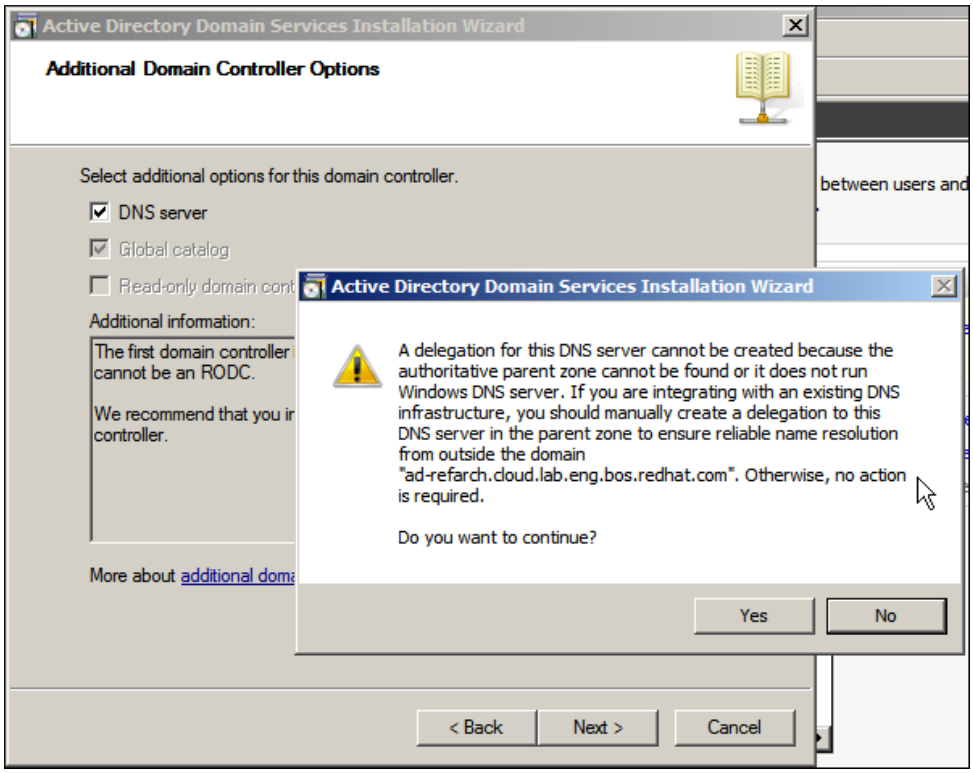


Figure B.3-11: Active Directory DNS Server 2

- Select the locations for the Active Directory domain controller database, log files and SYSVOL folders. The default locations are:

  Database folder: **C:\Windows\NTDS**
  Log files folder: **C:\Windows\NTDS**
  SYSVOL folder: **C:\Windows\SYSVOL**

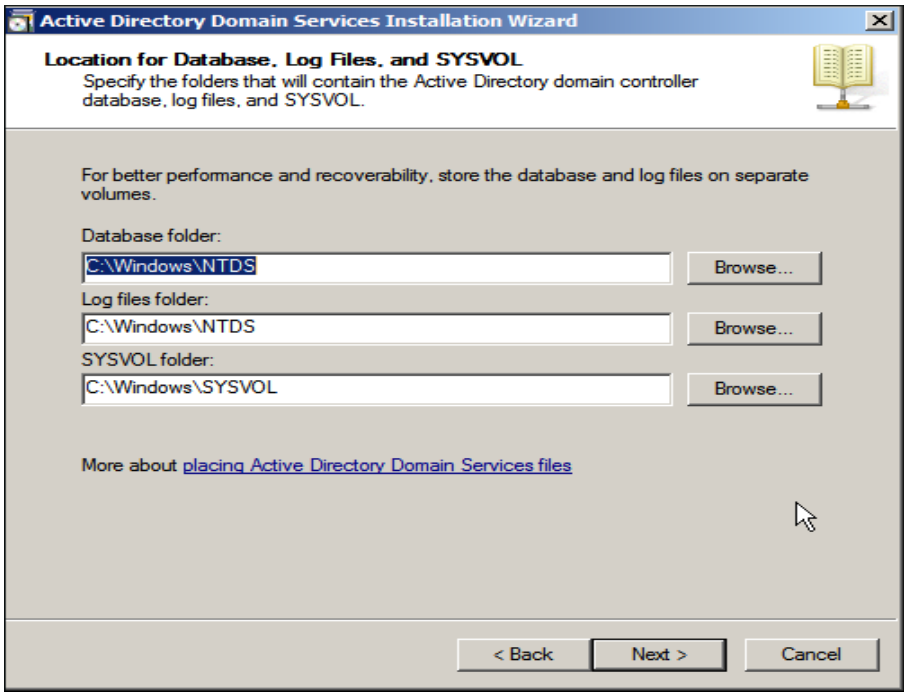


Figure B.3-12: Active Directory Folder Location

Note: For large installations each of these should be placed on separate volumes to maximize performance and recoverability

- Select **Next** to continue

Note: Unlike regular domain user passwords this password remains constant and must remain secure and confidential. This password should be complex and at least 7 characters long. It is highly recommended not to use the administrator's password and that it be securely stored.

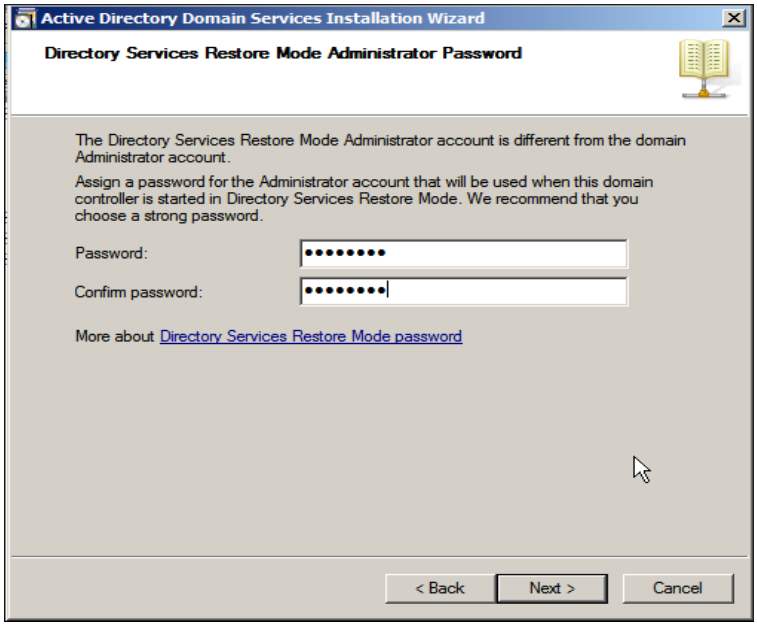


Figure B.3-13: Active Directory Administrator Password

- Select **Next** . After reviewing the Summary window select **Next**

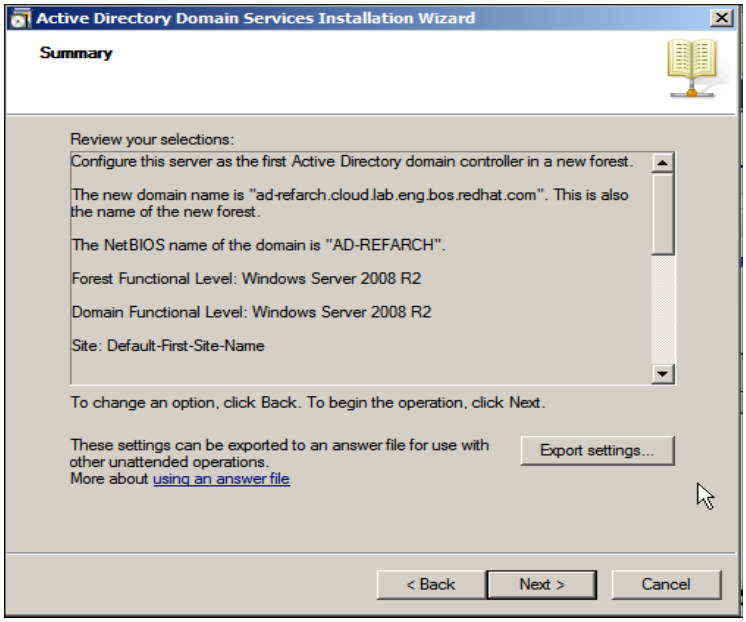


Figure B.3-14: Active Directory Configuration Summary

- After the wizard creates the Active Directory Domain select **Finish**
- Select **Reboot on completion** for changes to take effect.

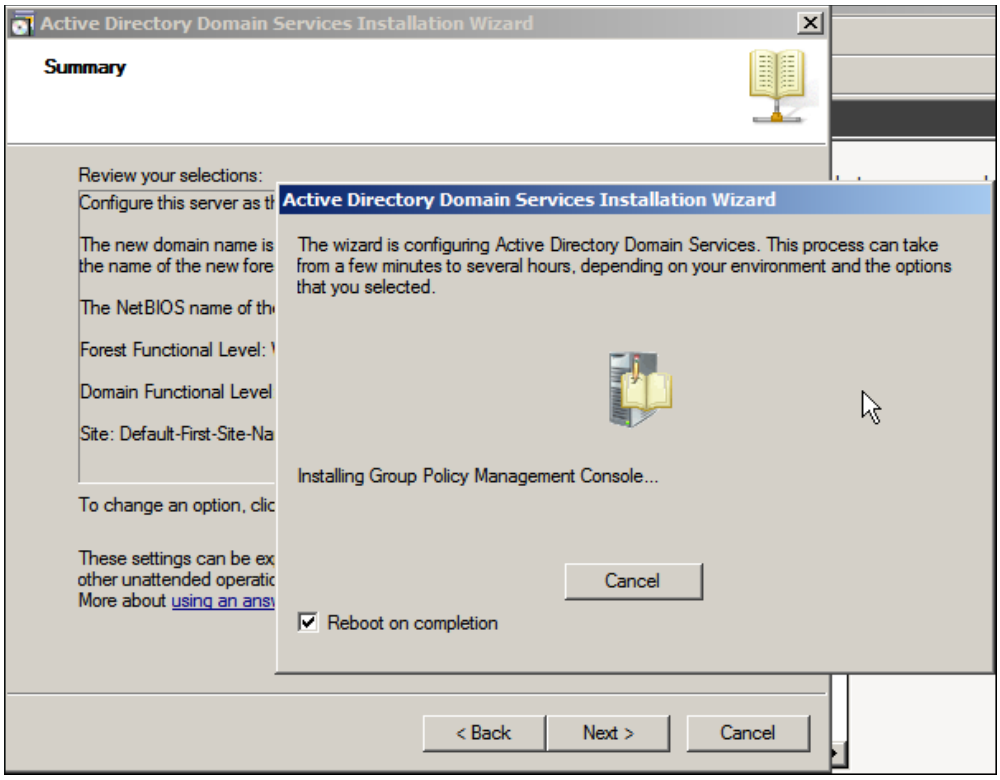


Figure B.3-15: Active Directory configuration Completion & Reboot

Note: From this point forward, the AD domain name (e.g. - ad-refarch) must be specified for all user logins

Figure B.3-16: Login with AD Domain

3. Configure Windows Time Service

   - From a Command Window (Start -> Run: cmd.exe) run:

     C:\AD-winsrv1>**w32tm /config /manualpeerlist:ntp1.xyz.redhat.com" /syncfromflags:manual /update**

   Note: Use the time server most appropriate to your environment

   - To verify, enter:

     C:\AD-winsrv1>**w32tm /query /status**
     Leap Indicator: 0(no warning)
     Stratum: 3 (secondary reference - syncd by (S)NTP)
     Precision: -6 (15.625ms per tick)
     Root Delay: 0.1095123s
     Root Dispersion: 0.0661492s
     ReferenceId: 0x0A10FF02 (source IP: 10.16.255.2)
     Last Successful Sync Time: 6/6/2013 3:25:43 PM
     Source: ns1.bos.redhat.com
     Poll Interval: 10 (1024s)

4. Create DNS Forward and Reverse Lookup Zones
    - Open *Server Manager* from the *Quick Launch toolbar*
    - Select **Roles -> DNS Server**
    - Expand **DNS Server** -> Expand **DNS** -> Expand computer name (***ad-winsrv1***)
    - Select **Configure a DNS Server**

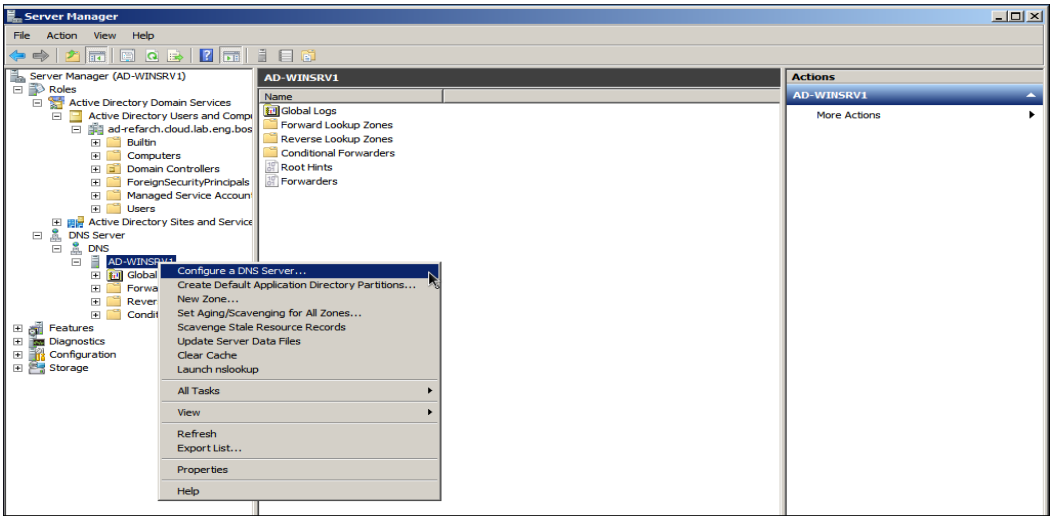


Figure B.3-17: DNS Server Configuration

- In the *Configure a DNS Server Wizard*, select **Create forward and reverse lookup zones** and select **Next**

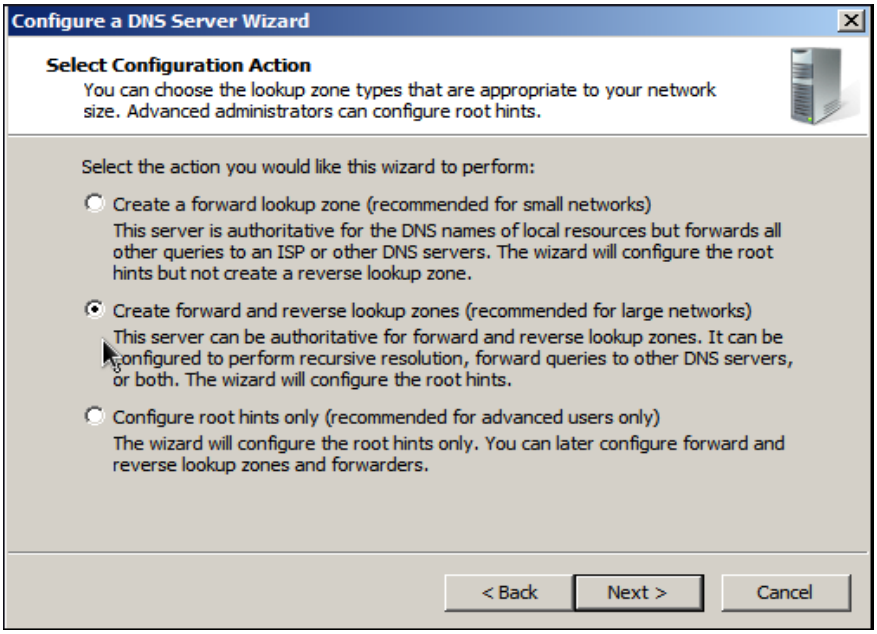


Figure B.3-18: DNS Server Configuration Action

- In the *Forward Lookup Zone* window, select **Yes, create a forward lookup zone now** and select **Next**

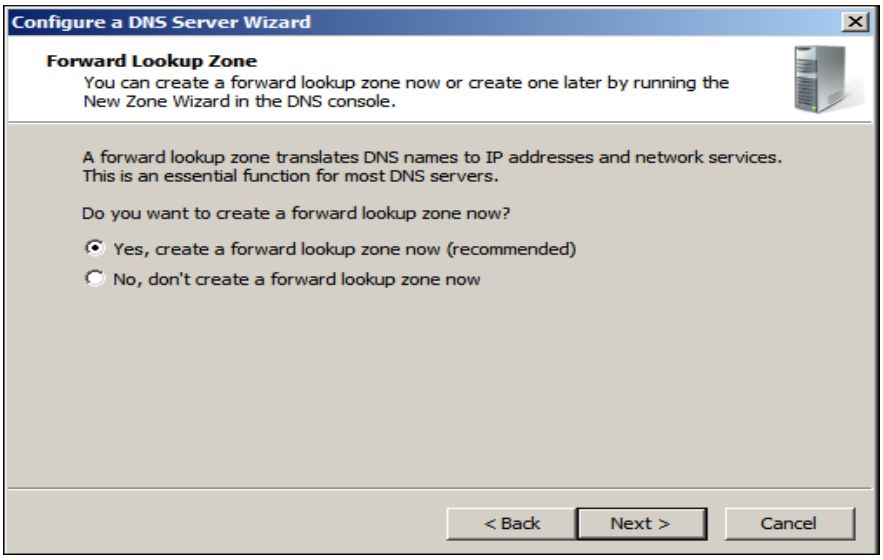


Figure B.3-19: DNS Server Forward Lookup Zone

- Select **Secondary zone** in the *Zone Type* window

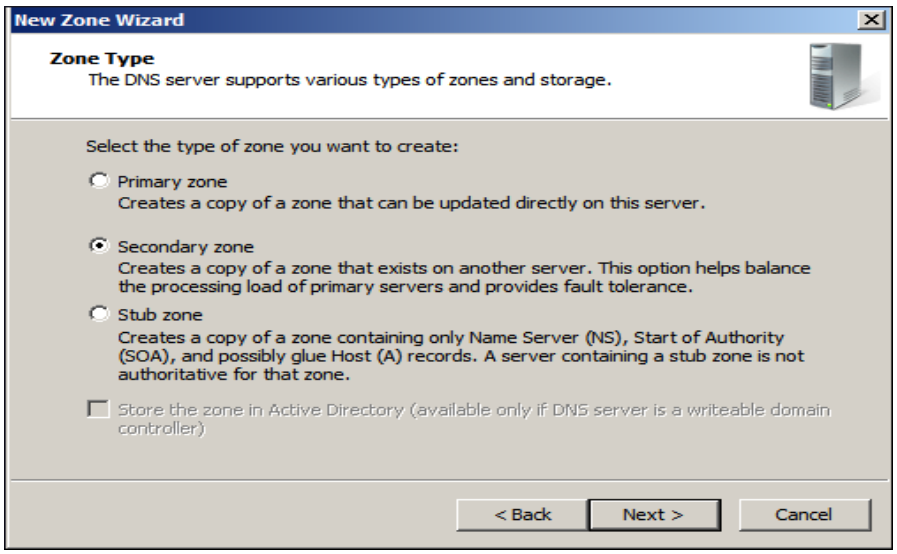


Figure B.3-20: DNS Server Secondary Zone 1

- Enter Zone name: **cloud.lab.eng.bos.redhat.com** and select **Next**

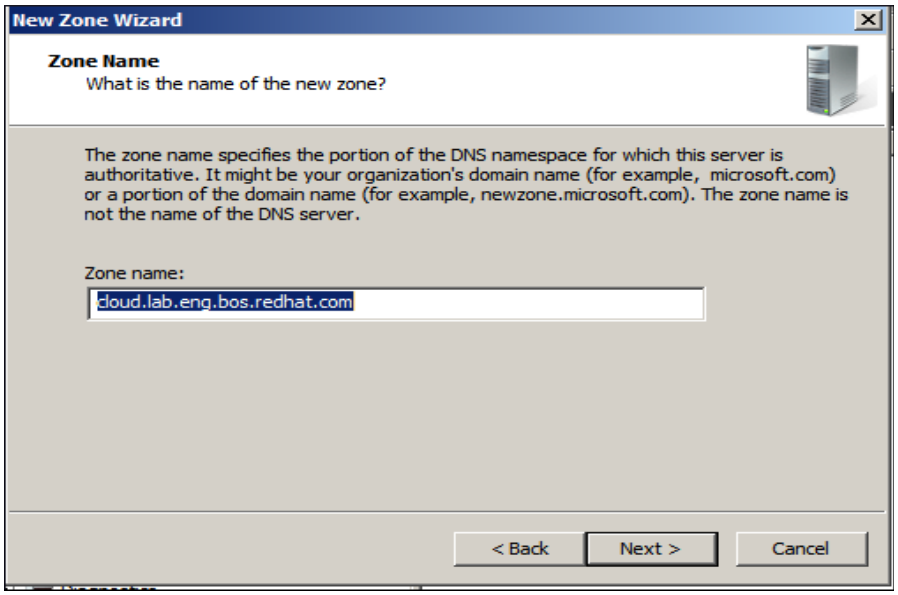


Figure B.3-21: DNS Server Secondary Zone 2

- Enter the DNS server information and select **Next**

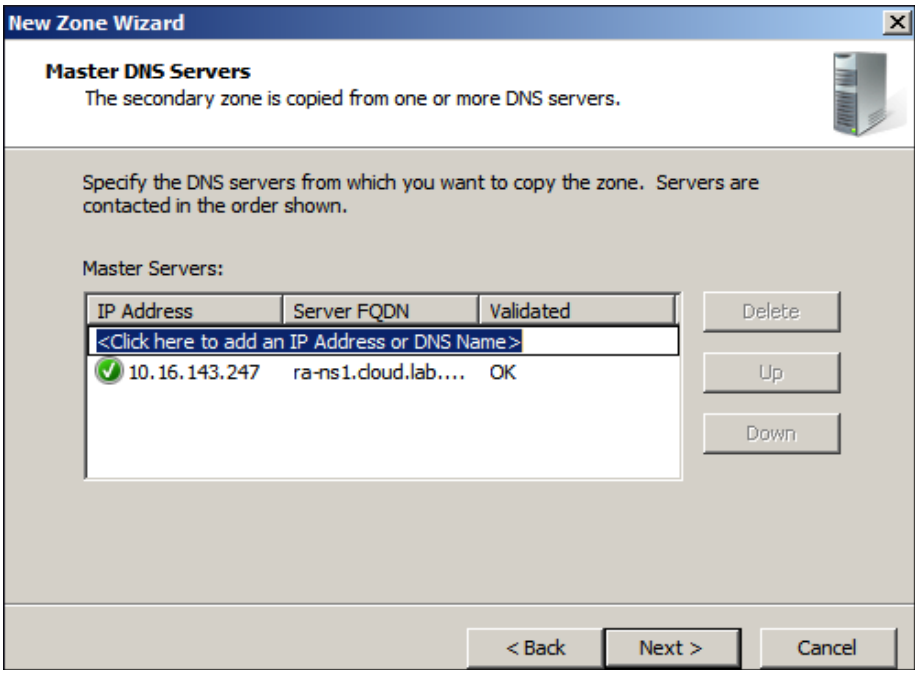


Figure B.3-22: DNS Server Master Server

- In the *Reverse Lookup Zone* window, select **Yes, create a reverse lookup zone now** and select **Next**

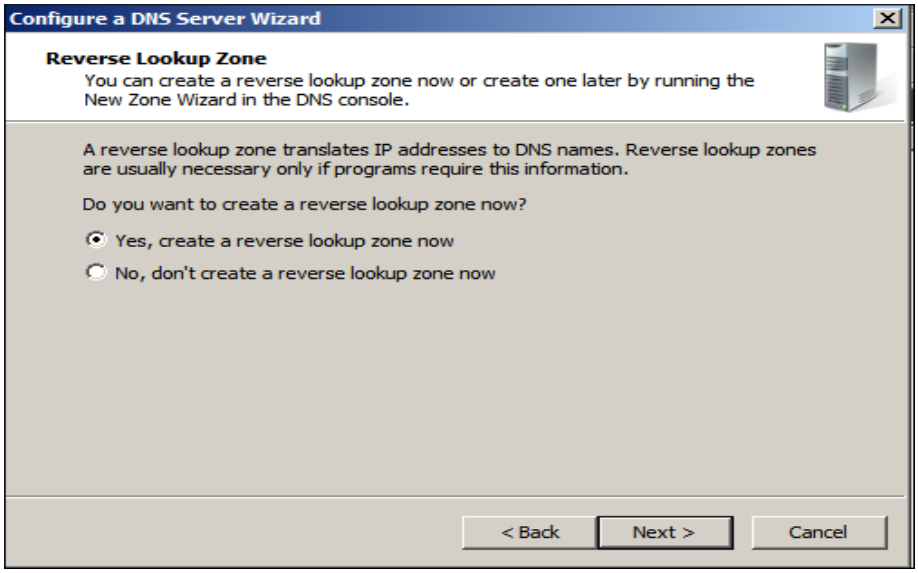


Figure B.3-23: DNS Server Secondary Zone Reverse Lookup Zone 1

- Select **Secondary zone** in the *Zone Type* window

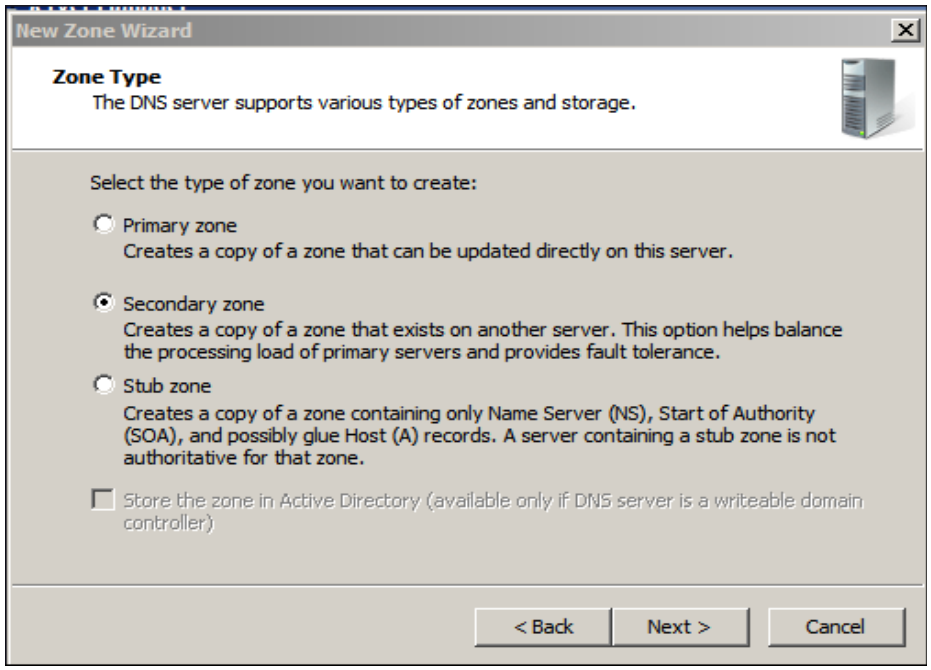


Figure B.3-24: DNS Server Secondary Zone Reverse Lookup Zone 2

- Select the appropriate Reverse lookup Zone

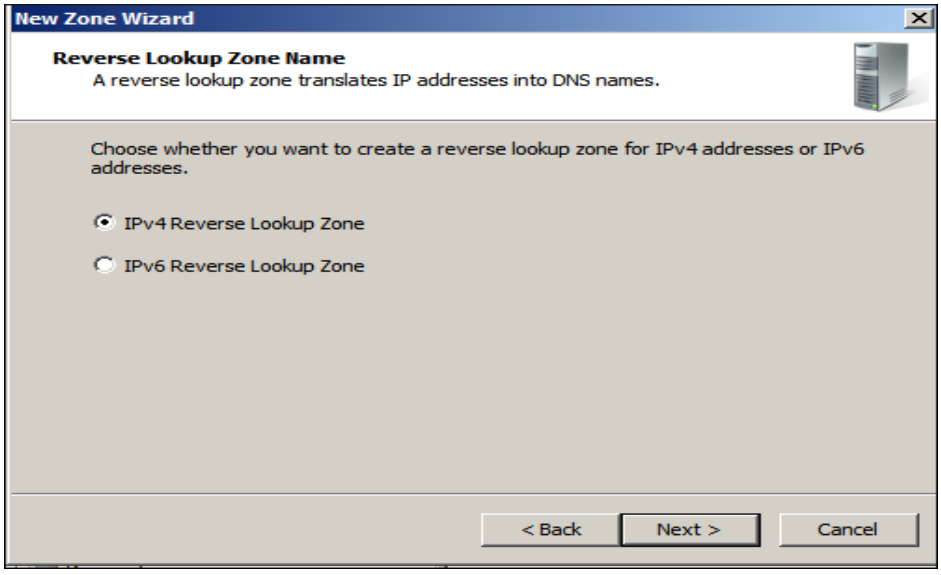


Figure B.3-25: DNS Server Secondary Zone Reverse Lookup Zone 3

- Enter the network ID of the zone **10.16.143**.

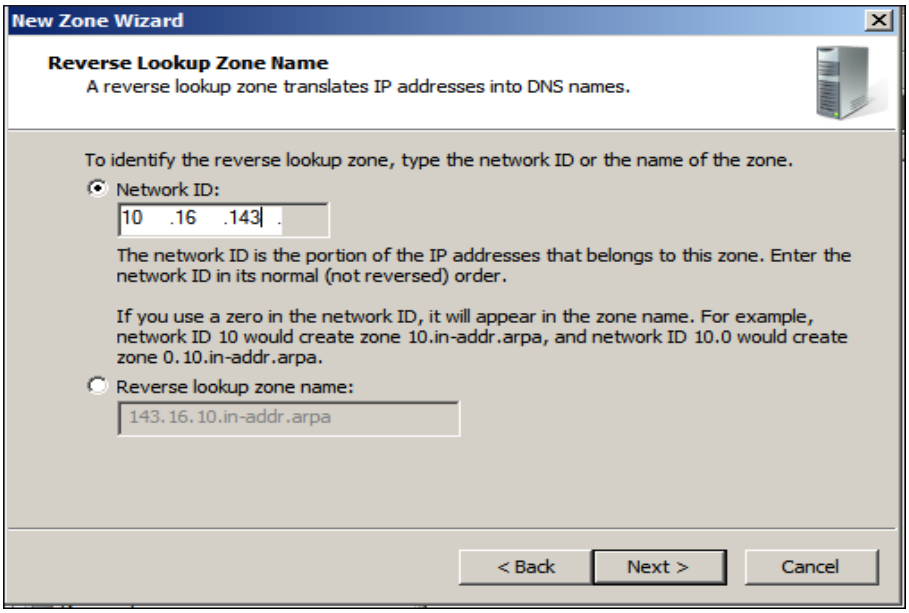


Figure B.3-26: DNS Server Secondary Zone Reverse Lookup Zone 4

- Enter the master DNS server details

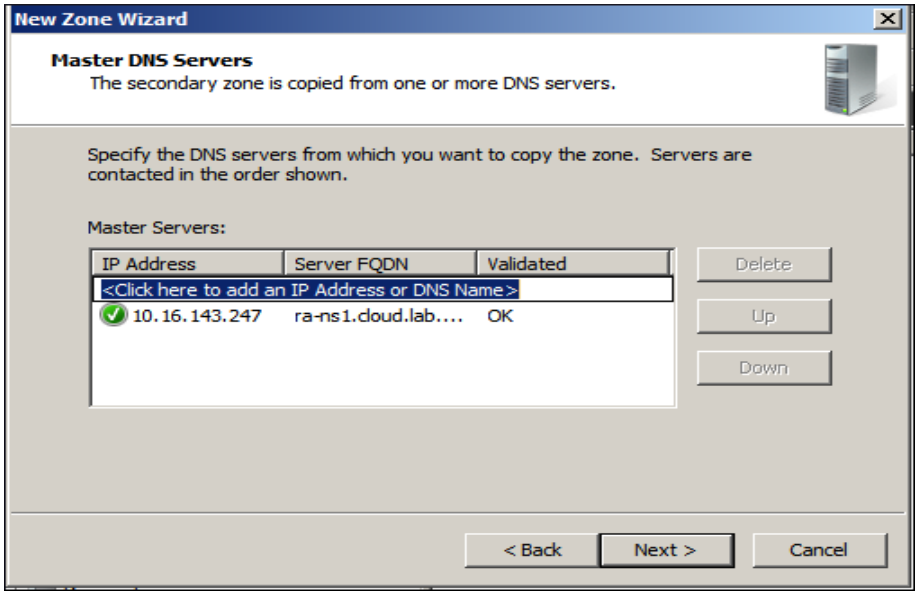


Figure B.3-27: DNS Server Secondary Zone Reverse Lookup Zone 5

- Select appropriate *Forwarder* selection

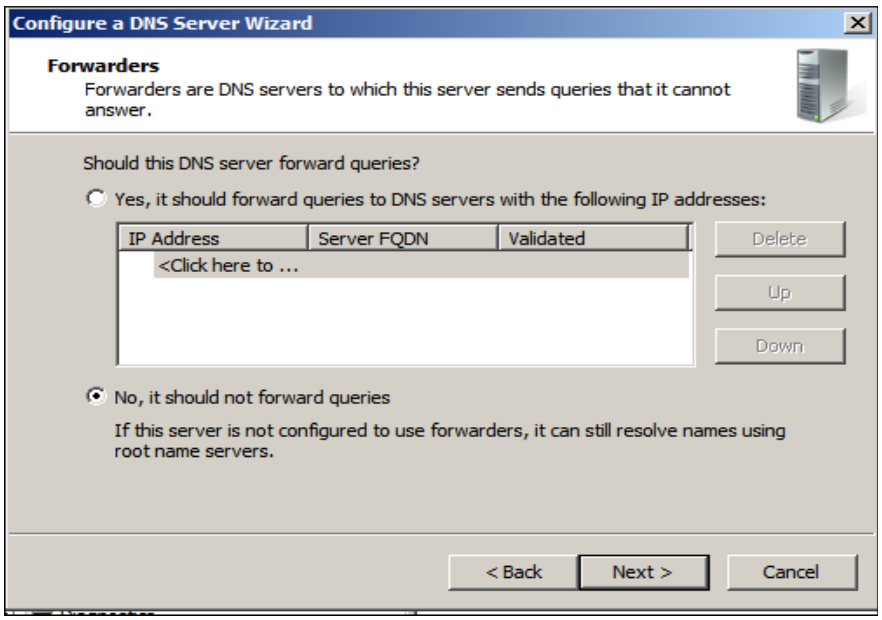


Figure B.3-28: Forward Queries

- Select **Finish** after successful completion of DNS Server configuration

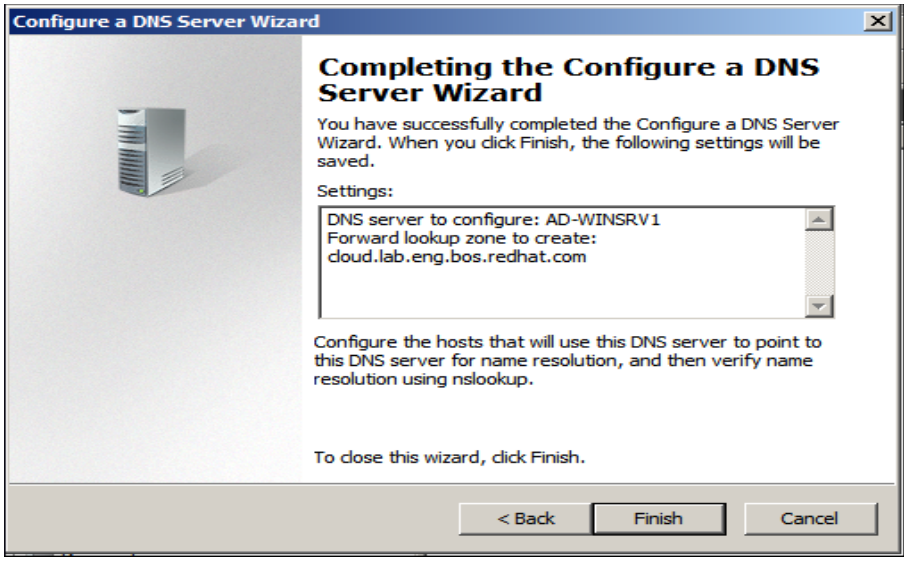


Figure B.3-29: DNS Server Configuration Summary

5. Restart DNS Service

   - Select **Configuration -> Services**
   - In the list of Services select **DNS Server**
   - Select **Restart the service**

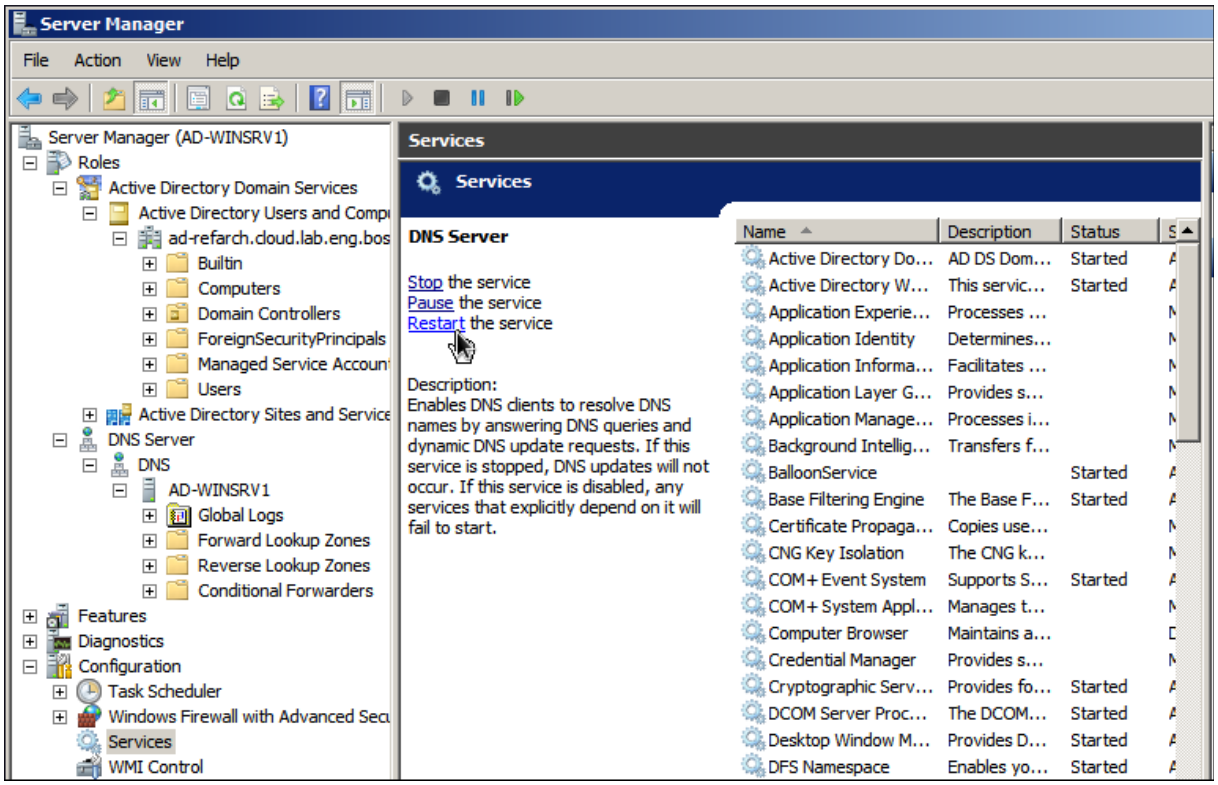


Figure B.3-30: Restart DNS Server

- Verify DNS is forwarding lookups. Open a Command Window and run:

```
C:\AD-WINSRV1>nslookup www.redhat.com
Server:  ra-ns1.cloud.lab.eng.bos.redhat.com
Address:  10.16.143.247

Non-authoritative answer:
Name:    e1890.b.akamaiedge.net
Address:  96.17.135.214
Aliases:  www.redhat.com
          wildcard.redhat.com.edgekey.net
          wildcard.redhat.com.edgekey.net.globalredir.akadns.net

C:\AD-WINSRV1>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : ad-winsrv1
   Primary Dns Suffix  . . . . . . . : ad-
refarch.cloud.lab.eng.bos.redhat
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : cloud.lab.eng.bos.redhat.com

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : cloud.lab.eng.bos.redhat.com
   Description . . . . . . . . . . . : Red Hat VirtIO Ethernet
Adapter
   Physical Address. . . . . . . . . : 00-1A-4A-10-88-09
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.16.136.53(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.248.0
   Default Gateway . . . . . . . . . : 10.16.143.254
   DNS Servers . . . . . . . . . . . : 10.16.143.247
                                       10.16.143.248
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

6. Verify Active Directory Domain Services

- Run the *Microsoft AD DS Best Practices Analyzer*:
- Select **Roles** -> **Active Directory Domain Services**
- Scroll down to the *Best Practices Analyzer*
- Select **Scan This Role**

  Review the results and correct any errors or warnings.

Note: The most common error is not having an NTP server set to synchronize time services. If this has not yet been done, follow the steps outlined in **Step 3. Configure Windows Time Service** before continuing.

If this has already been done, then synchronize/update by running the following:

```
C:\AD-WINSRV1> w32tm /config /computer:ad-winsrv1.ad-
refarch.cloud.lab.eng.bos.redhat.com /syncfromflags:domhier /update
```

- Open a Command Window (*Start* -> *Run*: cmd.exe) and run dcdiag:

```
C:\AD-WINSRV1> dcdiag
```

- If any errors are found, run the dcdiag in verbose mode for additional details:

```
C:\AD-WINSRV1> dcdiag /v
```

B.4 Create User Accounts

- Open *Server Manager* from the *Quick Launch toolbar*
- Select **Roles** -> **Active Directory Domain Services**
- Select **Active Directory Users and Computers**
- Open ***ad-refarch.cloud.lab.eng.bos.redhat.com*** (Domain)
- Right click on **Users**, select **New, User** and enter

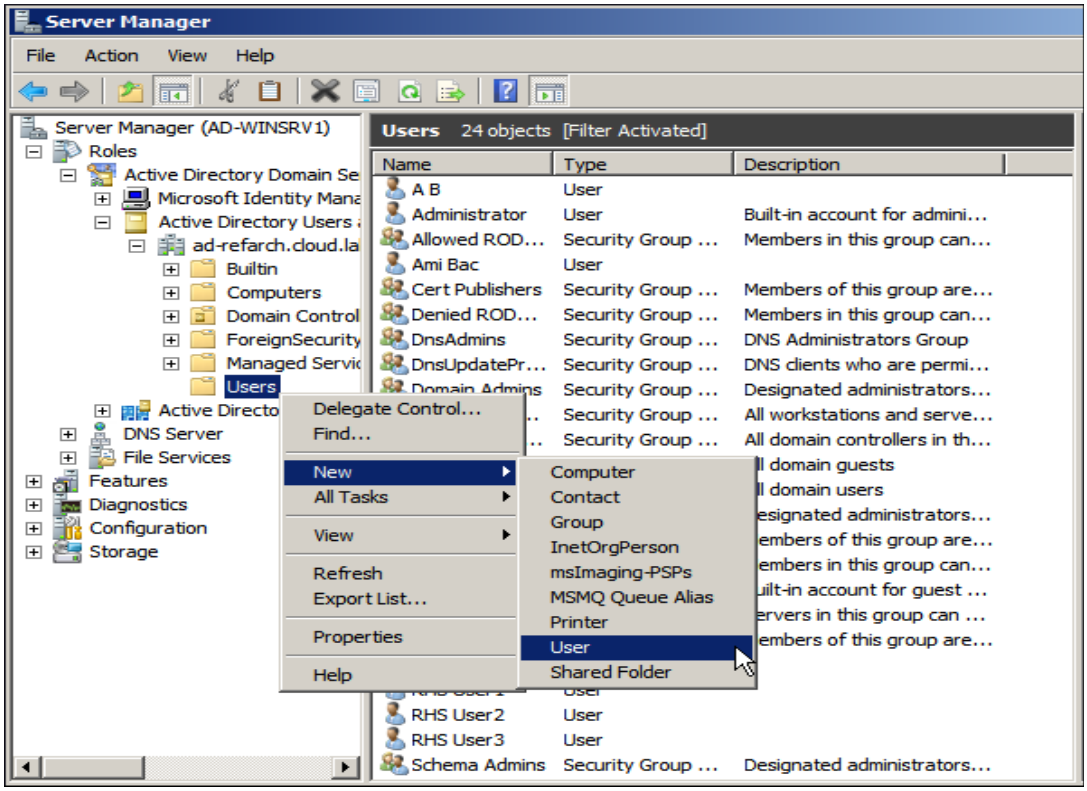


Figure B.4-1: New User 1

- Enter user information in the appropriate fields and select **Next**

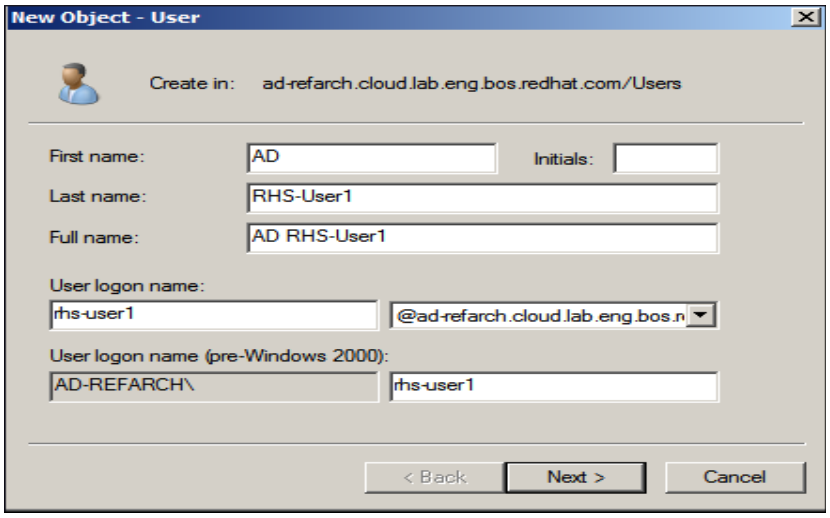


Figure B.4-2: New User 2

- Enter password and desired password policies and select Next

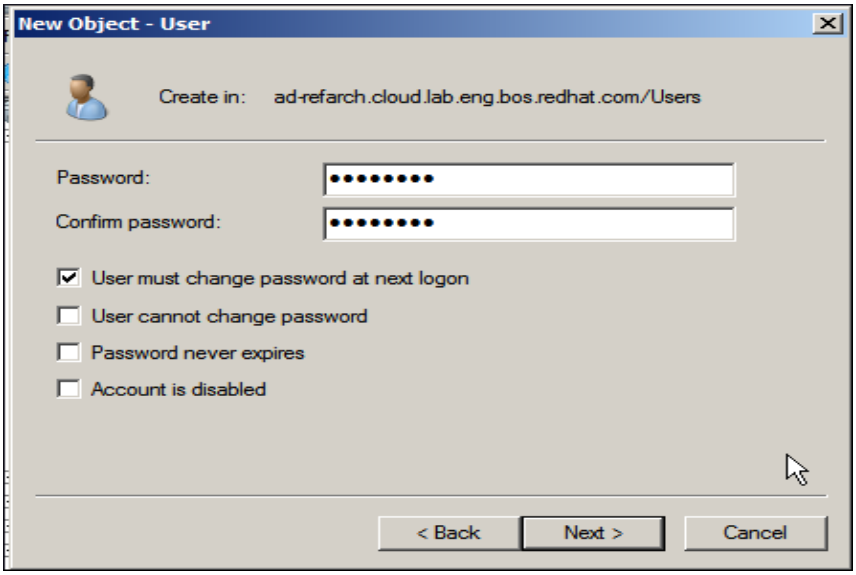


Figure B.4-3: User Credentials

- Select **Finish** after verifying the summary

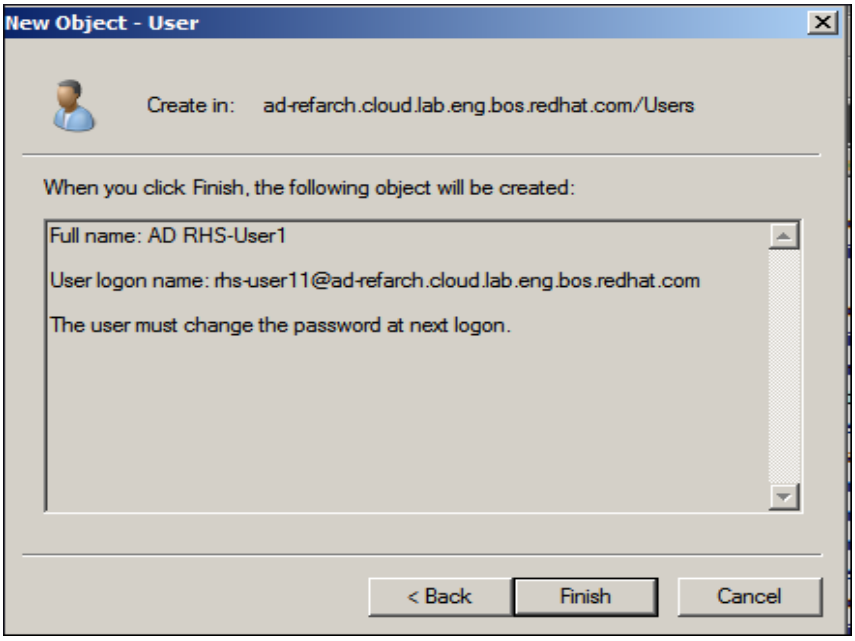


Figure B.4-4: New User Summary

Note: For more detail on Windows password policy requirements, see the following Microsoft TechNet article: http://technet.microsoft.com/en-us/library/cc736605.aspx

B.5 Add Red Hat Storage Server to AD Domain

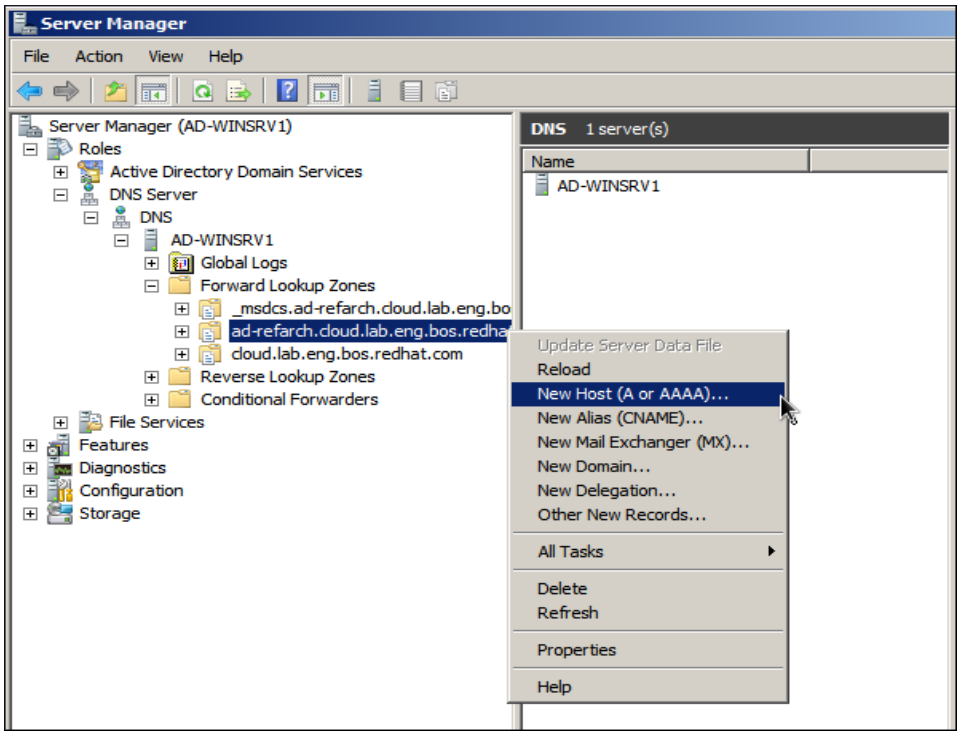


Figure B.4-5: Adding RHS Server to AD 1

Specify RHS server IP

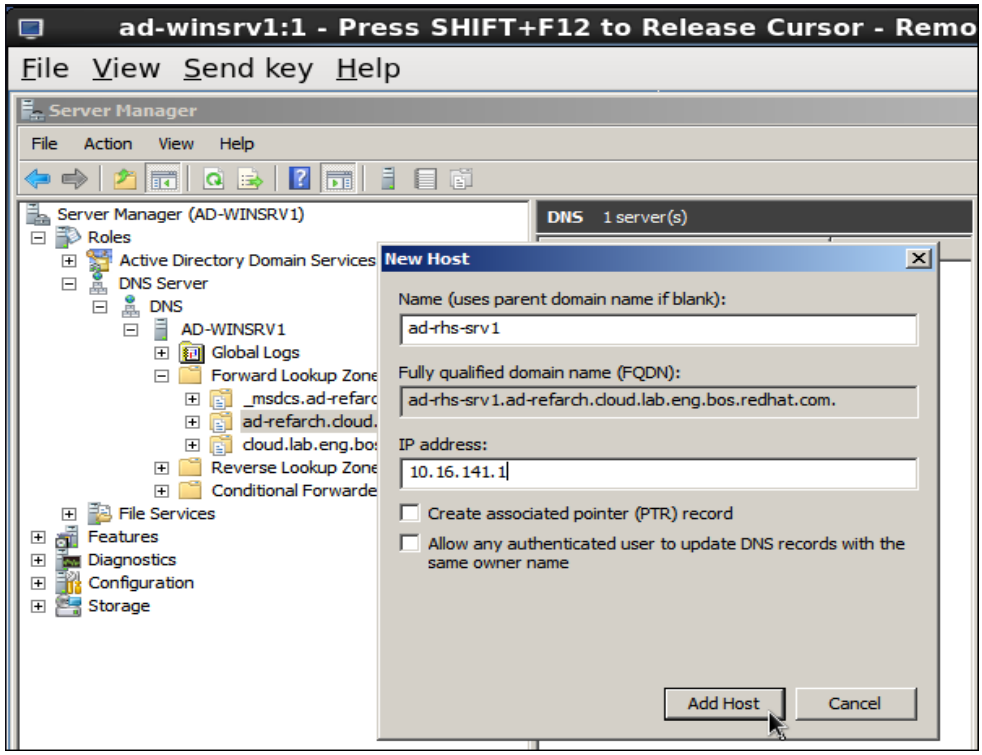


Figure B.4-6: Adding RHS Server to AD 2

Confirm host record

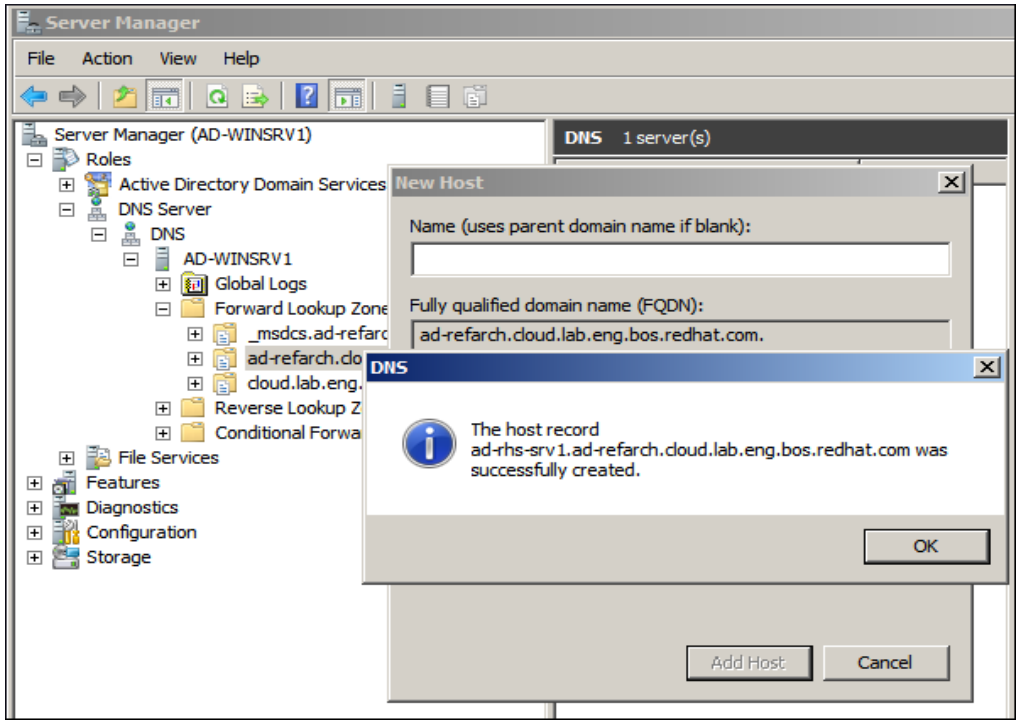


Figure B.4-7: Confirmation

Appendix C: Creating a Virtual Disk with Raid 6 using PERC Controller

1. During host system bootup, press <Ctrl><R> when the BIOS banner displays.

Figure 7-1: Launch PERC BIOS Configuration Utility

The *Virtual Disk Management* screen displays.

This will launch the PERC Integrated BIOS Configuration Utility. This utility will have the following three TABs on the top.

- VD Mgmt – Virtual Disk Management, which will be selected by default.
- PD Mgmt – Physical Disk Management
- Ctrl Mgmt – Controller Management

If there is more than one controller, the main menu screen displays. Select a controller, and press **Enter**. The *Virtual Disk Management* screen displays for the selected controller.

Note: PERC does not support creation of a virtual disk that combines SAS and SATA disks.

2. Use the arrow keys to highlight *Controller #* or *Disk Group #*. Press <F2> to display the actions you can perform.

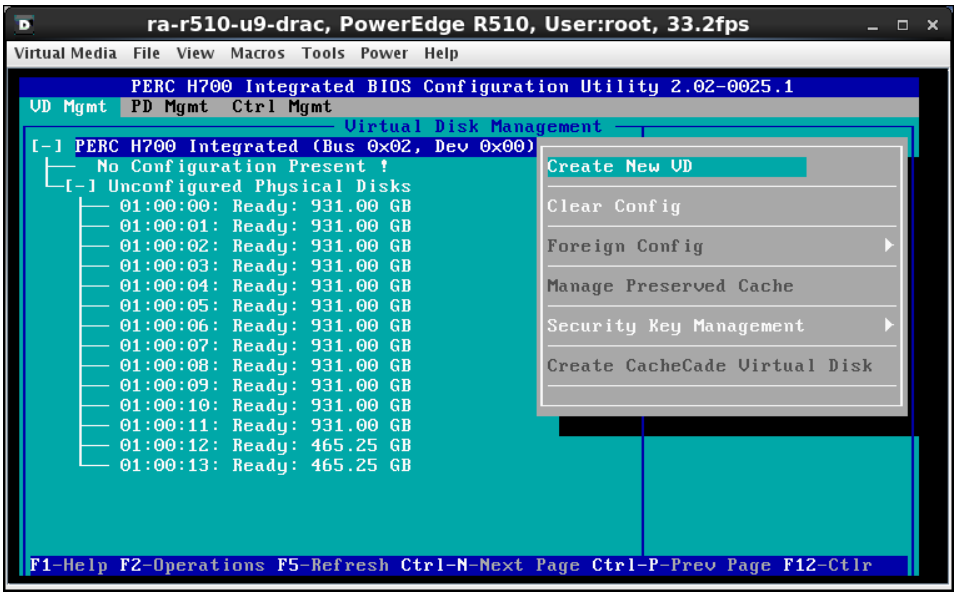


Figure C-1: Create Virtual Disk

4. Select *Create New VD* and press **Enter**. The *Create New VD* screen displays.

5. Press **Enter** to display the possible RAID levels, based on the physical disks available.
6. Press the down arrow key to select a RAID level (**RAID 6**)and press **Enter**.
7. Press <Tab> to move the cursor to the list of physical disks.
8. Use the arrow key to highlight a physical disk and press the spacebar, <Alt>, or **Enter** to select the disk.
9. Select additional disks, if desired.
10. Press <Tab> to move the cursor to the box *Basic Settings*.
11. Press <Tab> to access the **VD Size** field, and type a virtual disk name (**Vdisk2)**.

    The virtual disk size Vdisk2 displays the final pool size (***9310 GB***) based on the total disk space, number of disks and type of RAID selected.
12. Press <Tab> to move the cursor to *Advanced Settings.* Enter

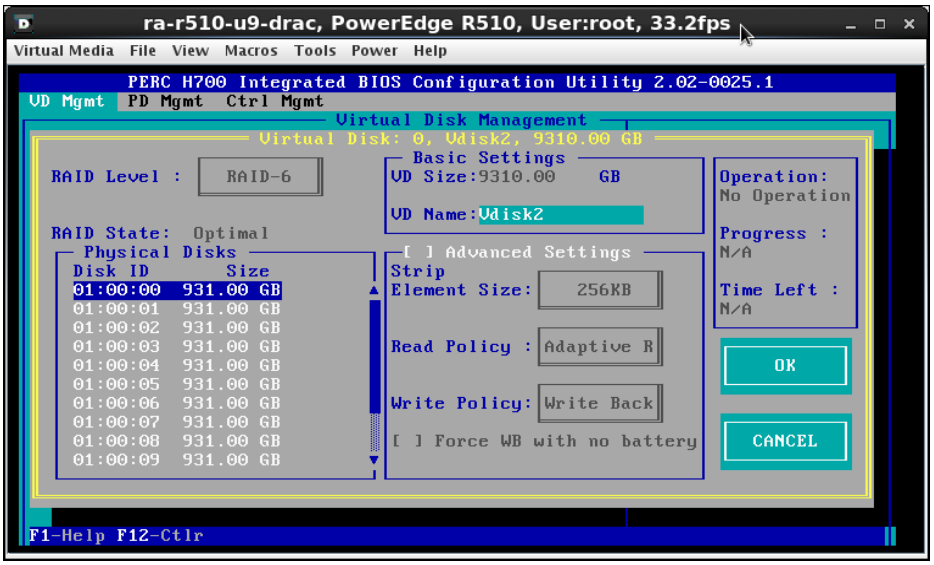


Figure C-2: Virtual Disk Settings

It is important that the following settings are selected in the *Advanced Settings*:

- Stripe Element Size: **256KB**
- Read Policy: **Adaptive**
- Write Policy: **Write Back**

Note: The Write Policy has to be set to Write Back for performance reasons. It is implied that the RAID Controller has *Battery Backup Unit BBU.* If this is not the case, Write Policy is set to *Write Through* by default and caching is disabled. Please verify with the vendor before proceeding further.

13. Select **OK** to accept the settings and press **Enter** to exit this window.
14. Initialize the Virtual Disk by selecting the right Virtual Disk on the VD Mgmt Tab -> **F2** -> **Initialization-> Start Init** (or Select **Fast Init**)

Appendix D: Red Hat Storage Server RHS2.1 Installation

Installing Red Hat Storage Server is very similar to installing Red Hat Enterprise Linux, but with
far fewer choices to make during the installation process. The basic procedure is outlined below.

1. Obtain an installation ISO for Red Hat Storage Server from https:// access.redhat.com The image used in this reference architecture is: **RHS-2.1-20130907.0-RHS-x86_64-DVD1.iso**
2. Mount the ISO image before the new hardware is booted up. In this case, the hardware is a Dell Server. The virtual CD/DVD can be mounted on the iDRAC console. Alternatively, this image can be burned as a physical DVD and used for installation.
3. Boot the server. The first screen will ask for the root password. Enter the same password
   twice then click on Next to continue.

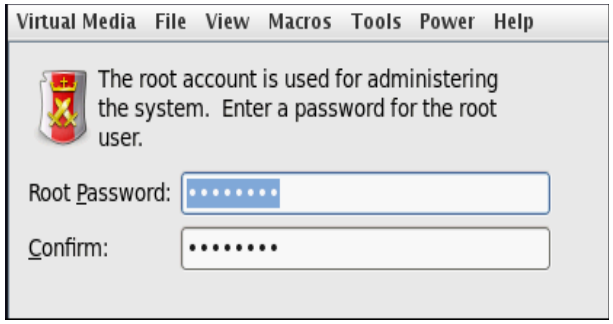


Figure D-1: RHS Installation-1

4. On the partitioning screen select one of the layout types to use. It is highly recommended to check the Review and modify partitioning layout check box.

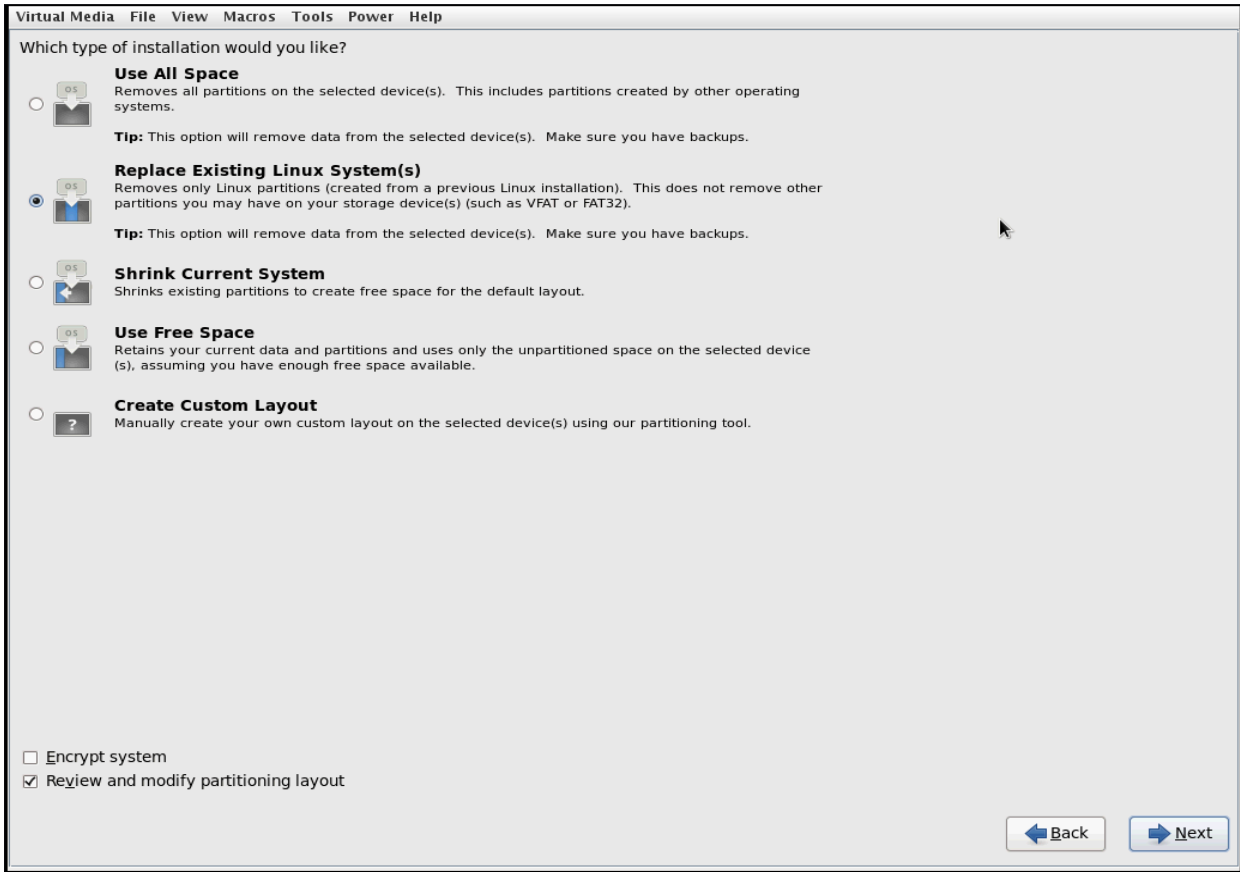


Figure D-2: RHS Installation-2

5. Available disks are displayed.

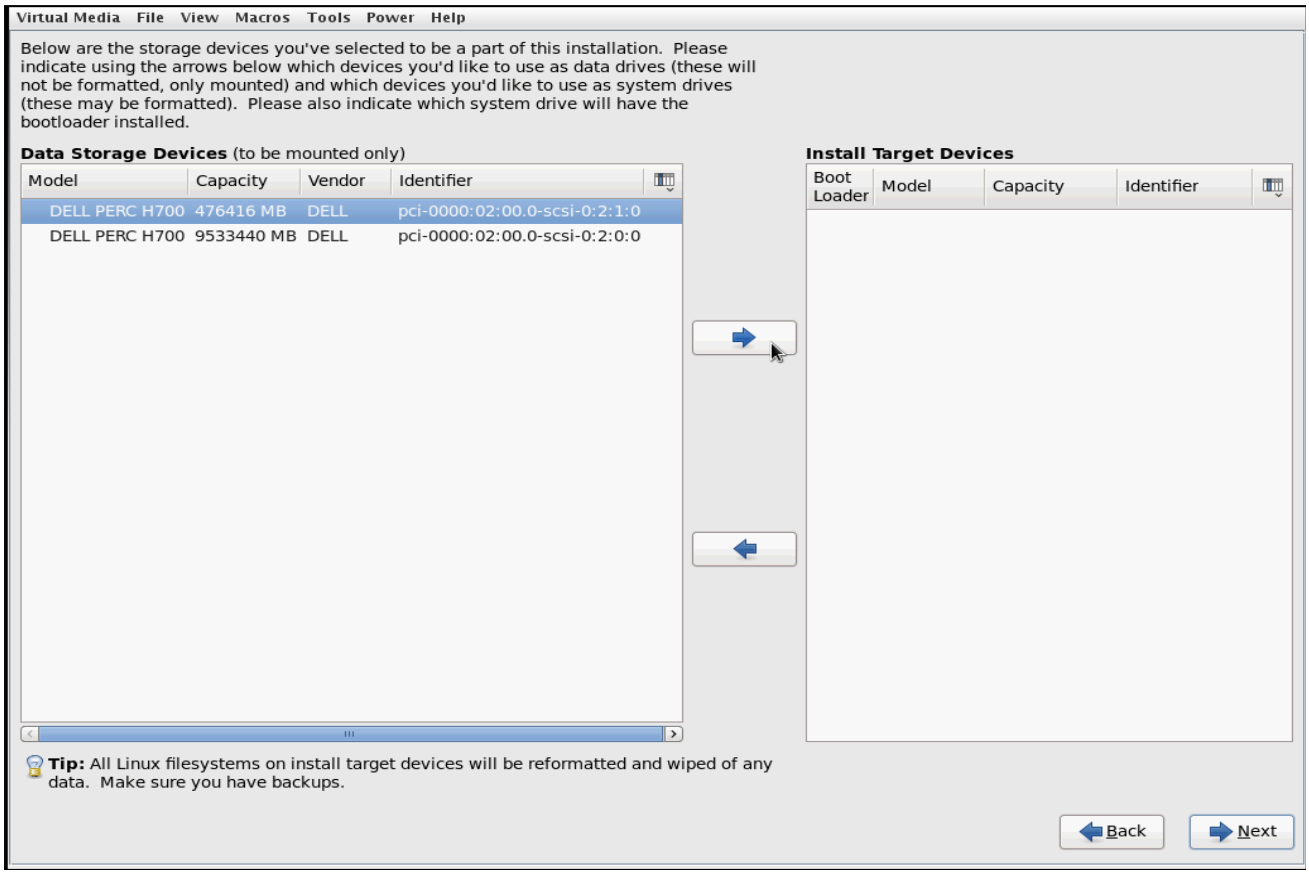


Figure D-3: RHS Installation - Disk selection 1

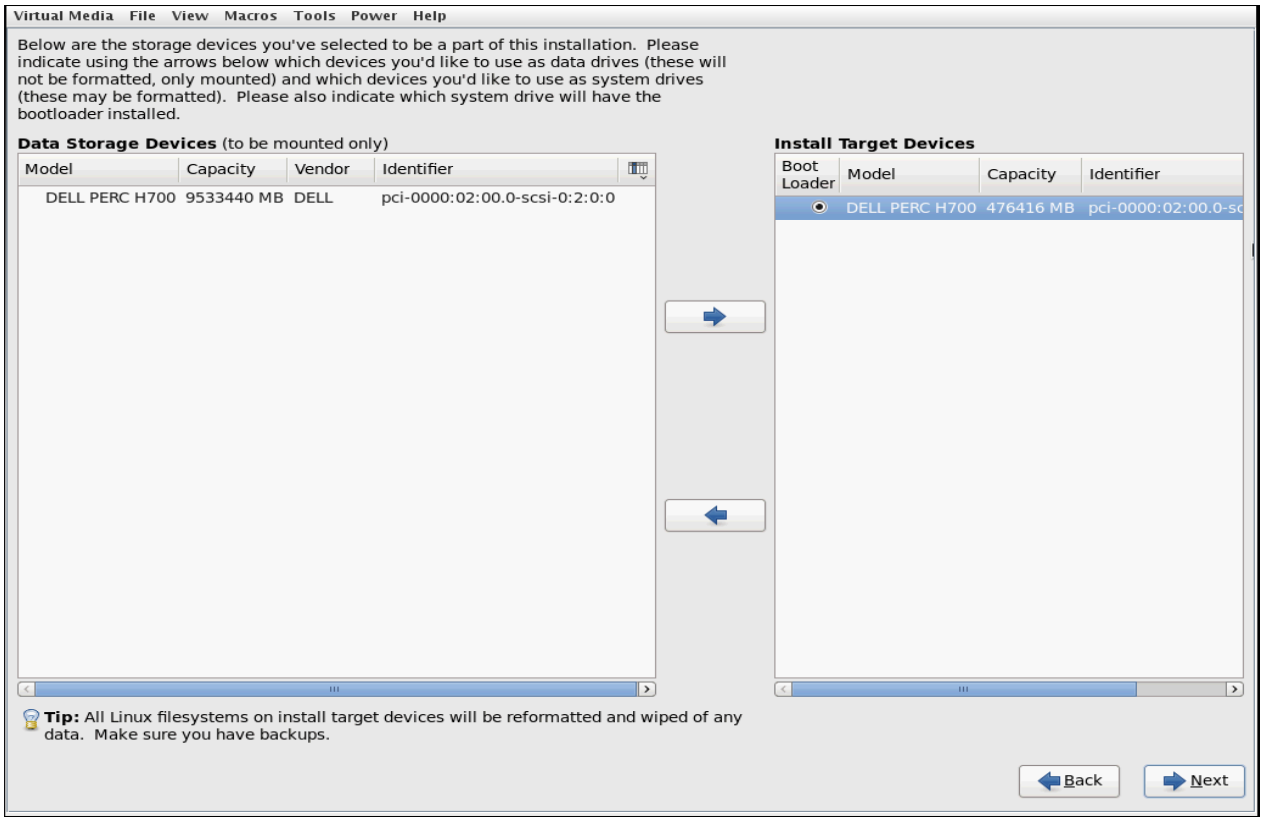


Figure D-4: RHS Installation - Disk selection 2

7. Ensure root partition has been allocated at least 2048 MB of disk space.

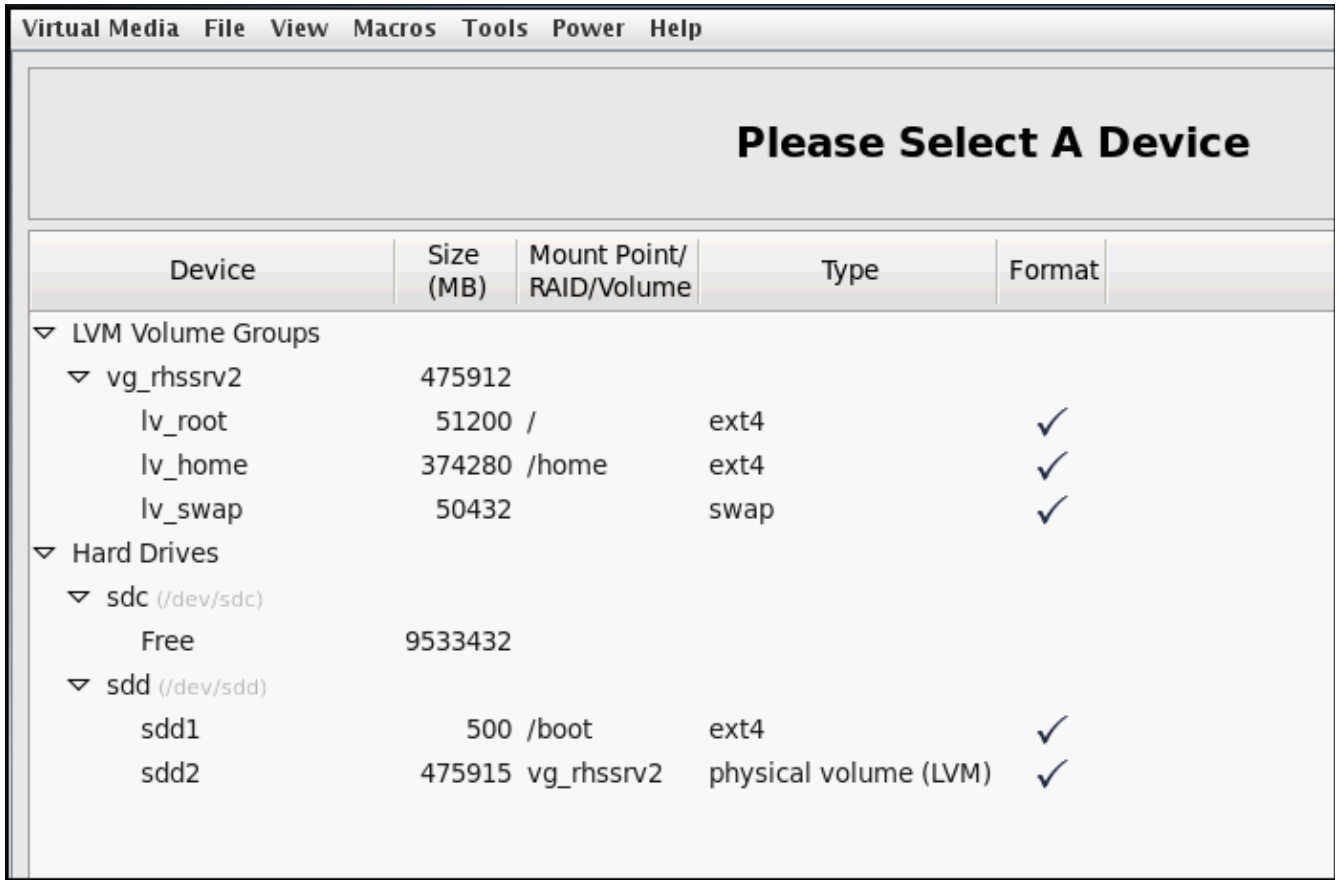


Figure D-5: RHS Installation - Root Filesystem

8. Confirm formatting of the right disks.

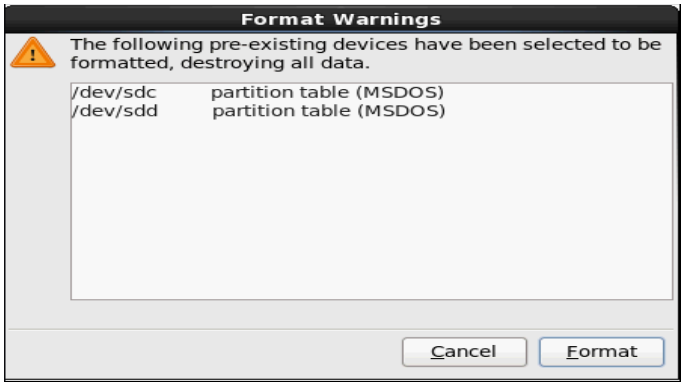


Figure D-6: RHS Installation - Disk formatting

9. Upon confirmation, the installer creates the system file systems and proceed to install the default package set for Red Hat Storage Server. By selecting **customize now** the software subsets can be viewed/modified.

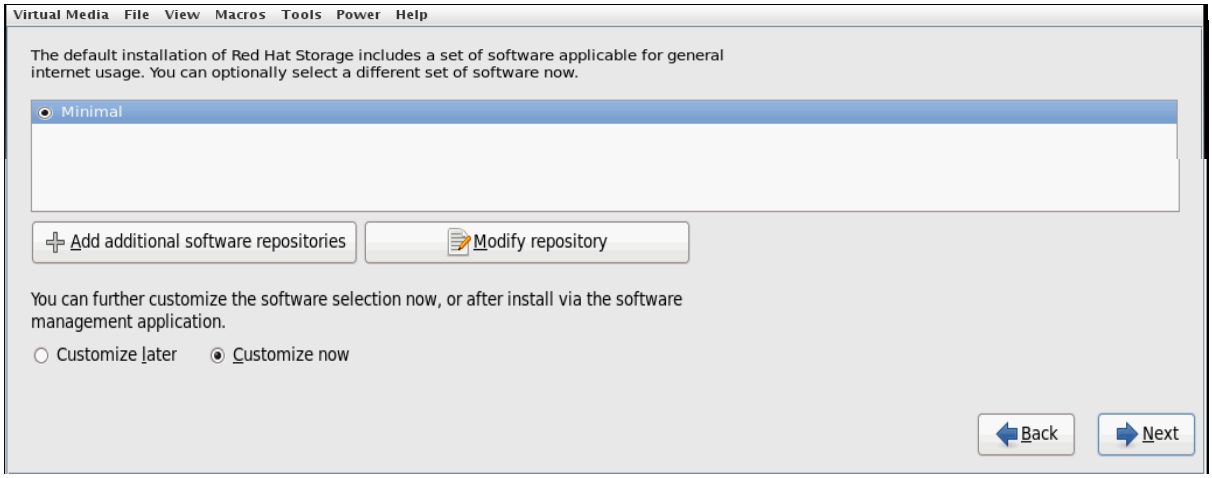


Figure D-7: RHS Installation - Software Selection 1

10. Select **Next** without having to deselect any of these subsets.

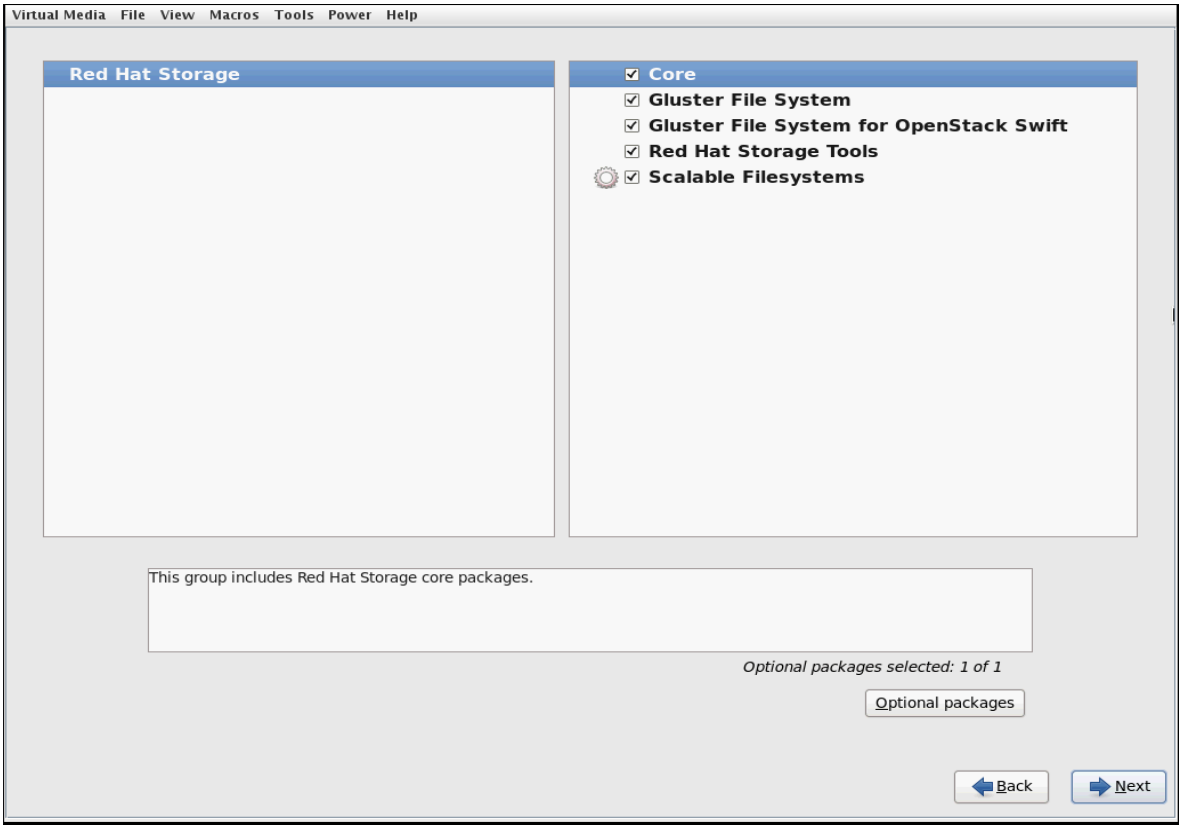


Figure D-8: RHS Installation - Software Selection 2

11. The installation begins.

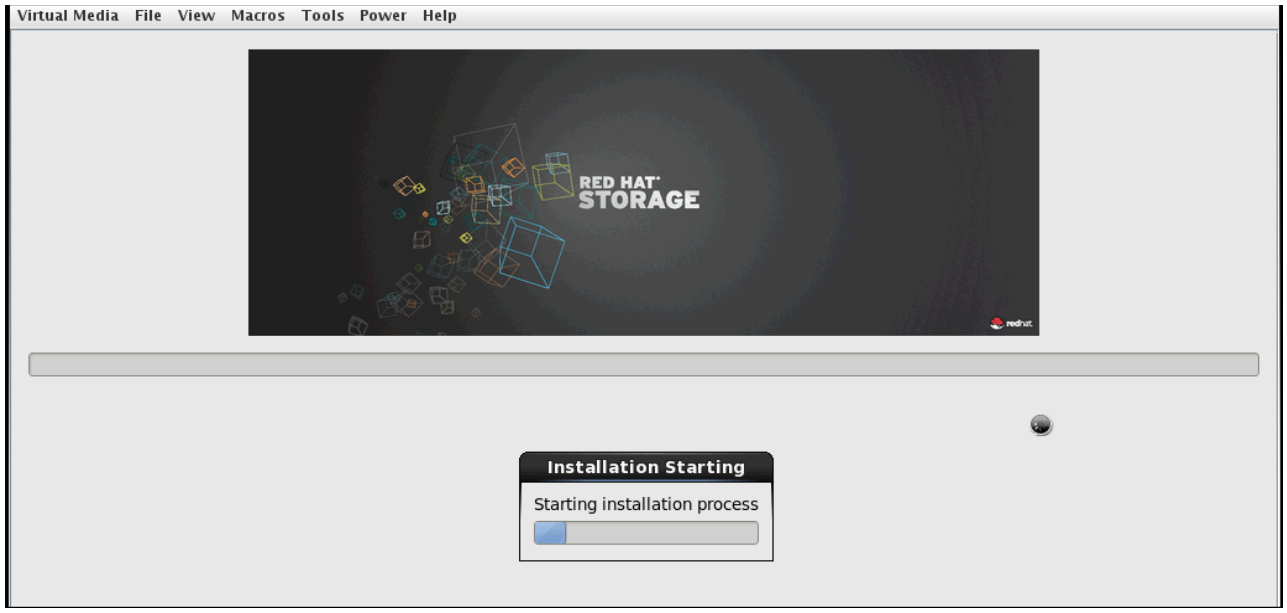


Figure D-9: RHS Installation - Final Step

12. Once the installer finishes, select **Reboot** button to finish the installation. After the server reboots, log in as root using the password specified earlier.
13. Verify if the glusterd service has been automatically started.

```
# service glusterd status
glusterd (pid  1434) is running...
```

Appendix E: Red Hat Storage Server Configuration – Performance Options Summary

The following tables describes the options/settings used to configure Red Hat Storage Environment. Some of the settings are based on recommendations from prior experiences, while the rest are customized to suit this setup in RHS 2.1 version.

Storage/Brick Configuration

Option/setting	Value	Description
Drives	12	Number of disks for Raid 6 setup for a brick
LUNS per brick	1	
Brick size & Count	-	Large but few bricks
Stripe size (kb)	256	Stripe size set to 256k instead of default 64k

Table E 1: Storage Options

LVM Configuration

Option/setting	Value	Description
Dataalignment (kb)	2560	Align the start of the data to a multiple of this number.

Table E 2: LVM Options

Example:

```
# pvcreate --dataalignment 2560k /dev/sdb
```

XFS Filesystem options

Option/setting	Value	Description
-i size (bytes)	512	Inode size of the filesystem
n size (bytes)	8192	Naming Size. Specifies the version and size parameters for the naming (directory) area of the filesystem in bytes.
su (kb)	256	Stripe unit size, matches with stripe unit of Raid
sw (count)	10	Usually the same as the number of stripe members/data disks in a RAID device

Table E 3: Filesystem Options

Example:

```
# mkfs.xfs -i size=512 -n size=8192 -d su=256k,sw=10 /dev/mapper/datavg-
rhsdata_lvol1
```

Mount options

Option/setting	Value	Description
inode location	inode64	Indicates that XFS is allowed to create inodes at any location in the filesystem
noatime	-	Access timestamps are not updated when a file is read. Improves disk performance.

Table E 4: Mount Options

Example:

/etc/fstab:

```
“/dev/mapper/datavg-rhsdata_lvol1  /rhs/storage1 xfs inode64, noatime 1 3”
```

Appendix F: Modification of hook scripts for Red Hat Storage Volumes

Starting with RHS version 2.1, the preferred way for creating an smb share of a Red Hat Storage Volume has changed. The previous method (RHS2.0) was to create a FUSE mount of the volume and share the mount point through samba. The new method eliminates the requirement of FUSE mount and changes in */etc/fstab* file. A glusterfs_vfs plug-in for samba was added, which makes a call to libgfapi to access the volume.

Since there is no longer a requirement for samba mount points in a standard scenario, automatic creation/deletion of samba mounts associated with a Red Hat Storage Volume start/stop has been removed. However in this case, the share directories under the Red Hat Storage Volumes require samba mount points to set access privileges. In order to restore the automated samba mount point feature, the following hook scripts were modified.

The addition/changes are highlighted below in bold:

/var/lib/glusterd/hooks/1/start/post/S30samba-start.sh

```
#!/bin/bash

#The preferred way of creating a smb share of a gluster volume has changed.
#The old method was to create a fu4se mount of the volume and share the
mount
#point through samba.
#New method eliminates the requirement of fuse mount and changes in fstab.
#glusterfs_vfs plugin for samba makes call to libgfapi to access the volume.
#This hook script automagically creates shares for volume on every volume
#start event by adding entries in smb.conf file & sending SIGHUP to samba.
##In smb.conf:
#glusterfs vfs plugin has to be specified as required vfs object.
#Path value is relative to the root of gluster volume;"/" signifies complete
#volume.

PROGNAME="Ssamba-start"
OPTSPEC="volname:"
VOL=
MNT_PRE="/mnt/samba"

function parse_args () {
        ARGS=$(getopt -l $OPTSPEC  -name $PROGNAME $@)
        eval set -- "$ARGS"

        while true; do
        case $1 in
        --volname)
         shift
         VOL=$1
         ;;
        *)
         shift
         break
```

```
         ;;
        esac
        shift
        done
}

function add_samba_share () {
        volname=$1
        STRING="\n[gluster-$volname]\n"
        STRING+="comment = For samba share of volume $volname\n"
        STRING+="vfs objects = glusterfs\n"
        STRING+="glusterfs:volume = $volname\n"
        STRING+="glusterfs:logfile = /var/log/samba/glusterfs-
$volname.log\n"
        STRING+="glusterfs:loglevel = 7\n"
        STRING+="path = /\n"
        STRING+="read only = no\n"
        STRING+="guest ok = yes\n"
        printf "$STRING"  >> /etc/samba/smb.conf
}

function sighup_samba () {
        pid=`cat /var/run/smbd.pid`
        if [ "$pid" != "" ]
        then
                kill -HUP "$pid";
        else
                /etc/init.d/smb condrestart
        fi
}

function get_smb () {
        volname=$1
        uservalue=

        usercifsvalue=$(grep user.cifs
/var/lib/glusterd/vols/"$volname"/info |\
                        cut -d"=" -f2)
        usersmbvalue=$(grep user.smb /var/lib/glusterd/vols/"$volname"/info
|\
                       cut -d"=" -f2)

        if [[ $usercifsvalue = "disable" || $usersmbvalue = "disable" ]];
then
                uservalue="disable"
        fi
        echo "$uservalue"
}

function add_fstab_entry () {
        volname=$1
        mntpt=$2
        mntent="`hostname`:/$volname $mntpt glusterfs
defaults,acl,transport=tcp 0 0"
        exists=`grep "$mntent" /etc/fstab`
```

```
        if [ "$exists" == "" ]
        then
            echo "$mntent" >> /etc/fstab
        fi
}

function mount_volume () {
        volname=$1
        mntpt=$2
        if [ "$(cat /proc/mounts | grep "$mntpt")" == "" ]; then
                mount -t glusterfs `hostname`:$volname $mntpt && \
                                add_fstab_entry $volname $mntpt
        fi
parse_args $@
if [ $(get_smb "$VOL") = "disable" ]; then
        exit 0
fi

add_samba_share $VOL
add_fstab_entry
mkdir -p $MNT_PRE/$VOL
sleep 5
mount_volume $VOL $MNT_PRE/$VOL

sighup_samba
```

/var/lib/glusterd/hooks/1/stop/S30samba-stop.sh

```
#! /bin/bash

#The preferred way of creating a smb share of a gluster volume has changed.
#The old method was to create a fuse mount of the volume and share the mount
#point through samba.
#
#New method eliminates the requirement of fuse mount and changes in fstab.
#glusterfs_vfs plugin for samba makes call to libgfapi to access the volume.
#
#This hook script automagically removes shares for volume on every volume
stop
#event by removing the volume related entries(if any) in smb.conf file.

PROGNAME="Ssamba-stop"
OPTSPEC="volname:"
VOL=
MNT_PRE="/mnt/samba"

function parse_args () {
        ARGS=$(getopt -l $OPTSPEC  -name $PROGNAME $@)
        eval set -- "$ARGS"

        while true; do
        case $1 in
        --volname)
         shift
         VOL=$1
```

```
         ;;
        *)
         shift
         break
         ;;
        esac
        shift
        done
}

function del_samba_share () {
        volname=$1
        cp /etc/samba/smb.conf /tmp/smb.conf
        sed -i "/gluster-$volname/,/^$/d" /tmp/smb.conf &&\
                cp /tmp/smb.conf /etc/samba/smb.conf
}

function umount_volume () {
        volname=$1
        mnt_pre=$2
        umount -l $mnt_pre/$volname
}

function remove_fstab_entry () {
        volname=$1
        mntpt=$2
        mntent="`hostname`:/$volname $mntpt glusterfs defaults,transport=tcp
0 0"
        esc_mntent=$(echo -e "$mntent" | sed 's/\//\\\//g')
        exists=`grep "$mntent" /etc/fstab`
        if [ "$exists" != " " ]
        then
                sed -i /"$esc_mntent"/d /etc/fstab
        fi
}

function sighup_samba () {
        pid=`cat /var/run/smbd.pid`
        if [ $pid != "" ]
        then
                kill -HUP $pid;
        else
                /etc/init.d/smb condrestart
        fi
}

parse_args $@
del_samba_share $VOL
umount_volume $VOL $MNT_PRE
remove_fstab_entry $VOL $MNT_PRE/$VOL
sighup_samba
```

Appendix G: Red Hat Storage Server Initialization Script (optional)

The following script example created by Veda Shankar is displayed below. This script (run after RHS ISO image installation), performs various steps as mentioned under **Section 4 Deploy Red Hat Storage Server Infrastructure**:

At a high level, the script automates the following steps:

- rhn_register and update
- Identifies the data disk and performs LVM functions
- Creates XFS filesystem with desired settings and mounts it.
- Executes performance tuning using Tuned
- Creates bricks

Note: This has been provided as a reference for the convenience of the users. This must be customized and validated before use.

```
#!/bin/bash
# Title : rhs-system-init.sh
# Author : Veda Shankar
# Description :
# RHS system initialization script.  The script is supposed to be run after
# ISO installtion and setting up the network.
# The script does the following:
#     - Identify the RAID volume using the WWID and create a brick.
#     - Based on the use case, run the corresponding performance tuning
profile.
#     - Register with RHN and subscribe to correct software channels.
#     - Run yum update to apply the latest updates.
#
# History:
# 12/13/2012 Veda Shankar  Created
# 12/20/2012 Veda : Check the correct RHN channels before applying updates
# 01/18/2013 Veda : Incorporate recommended options for pv create, mkfs and
#                   mounting.
# 01/24/2013 Veda : Provide -n option for dry-run.
# 04/15/2013 Veda : Additional performance options for mkfs and fstab.
# 05/01/2013 Veda : Use lvmdiskscan to detect disks
#                   Auto detect whether physical or virtual.
#                   Changed the inode size to 512 for object use case.
#                   Set the performance profile to rhs-virtualization for
#                   virtual workload.
# 05/22/2013 Veda : Made sure that mkfs and fstab mount options follow the
#                   latest recommendations for XFS parameters for RHS
bricks.
# 08/27/2013 Veda : Provide -r option to skip RHN registration.
# 08/27/2013 Veda : Register with RHS 2.1 and RHEL 6.4.z channels.
#
```

```
# Default settings
ME=$(basename $0);
dryrun=0
skip_registration=0
logfile=/root/rhs-init.log
vgname_base=RHS_vg
lvname_base=RHS_lv
brickpath=/rhs
workload=general
inode_size=512
tune_profile=rhs-high-throughput

# LVM settings best suited for the standard RHS deployment
# which has 12 drives in RAID6 configuration with 256-KB stripe size.
stripesize=256k
stripe_elements=10
dataalign=2560k
fs_block_size=8192
percent_inodes=15
xfs_alloc_groups=64

# Disk path name variables
disk_path=NODISK
root_disk_path=NODISK

#exec > >(tee /root/rhs-init.log)
#exec 2>&1

function usage {
    cat <<EOF
Usage:  $ME [-h] [-u virtual|object]

General:
  -u <workload>   virtual - RHS is used for storing virtual machine images.
                  object  - Object Access (HTTP) is the primary method to
                            access the RHS volume.
                  The workload helps decide what performance tuning profile
                  to apply and other customizations.
                  By default, the  general purpose workload is assumed.

  -n              Dry run to show what devices will be used for brick
creation.
  -r              Skip RHN Registration.
  -h              Display this help.

EOF
    exit 1
}


function quit {
    exit $1
}

function yesno {
```

```
   while true; do
       read -p "$1 " yn
       case $yn in
           [Yy]* ) return 0;;
           [Nn]* ) return 1;;
           * ) echo "Please answer yes or no.";;
       esac
   done
}


function create_pv {
    dev=$1
    pvdisplay | grep -wq $dev
    ret=$?
    if [ $ret -eq 0 ]
    then
        echo "$dev Physical Volume exits!"
        return 1
    fi
    echo "Create Physical Volume with device $dev"
    [ $dryrun -eq 1 ] && return 0

    pvcreate --dataalignment $dataalign $dev
    return  $?
}

function create_vg {
    dev=$1
    vgname=$2
    echo "Create Volume Group $vgname."
    [ $dryrun -eq 1 ] && return 0

    vgcreate $vgname $dev
    return  $?
}

function create_lv {
    vgname=$1
    lvname=$2
    echo "Create Logical Volume $lvname."
    [ $dryrun -eq 1 ] && return 0

    # lvcreate -l 85%FREE -n $lvname $vgname
    lvcreate -l 100%FREE -n $lvname $vgname
    return  $?
}

function create_fs {
    vgname=$1
    lvname=$2
    echo "Create XFS file system /dev/$vgname/$lvname."
    echo "mkfs.xfs -i size=$inode_size -n size=$fs_block_size -d
su=$stripesize,sw=$stripe_elements  /dev/$vgname/$lvname"
    [ $dryrun -eq 1 ] && return 0
```

```
    mkfs.xfs -i size=$inode_size -n size=$fs_block_size -d
su=$stripesize,sw=$stripe_elements /dev/$vgname/$lvname
    return  $?
}

function create_fstab_entry {
    [ $dryrun -eq 1 ] && return 0
    vgname=$1
    lvname=$2
    mount=$3
    uuid=`xfs_admin -u /dev/$vgname/$lvname | cut -f3 -d " "`
    echo $uuid
    grep -wq $uuid /etc/fstab > /dev/null 2>&1 && return 1
echo "Create fstab entry for /dev/mapper/$vgname-$lvname ($uuid)."
    echo "UUID=$uuid $mount xfs \
          inode64,noatime,nodiratime 1 2" >> /etc/fstab
    return 0
}

function create_bricks {

    declare -a device_name=(`lvmdiskscan | grep $disk_path \
                              | grep -v $root_disk_path | awk '{ print
$1 }'`)
    echo "brick devices:"
    echo ${device_name[*]}
    count=0
    dev_count=1
    for dev in "${device_name[@]}"
    do
       echo "---- Device# ${dev_count} ----"
       vgname=$vgname_base$dev_count
       lvname=$lvname_base$dev_count

       # Create Physical Volume
       create_pv $dev || exit 1

       # Create Volume Group
       create_vg $dev $vgname || exit 1

       # Create Logical Group
       create_lv $vgname $lvname || exit 1

       # Create XFS file system
       create_fs $vgname $lvname || exit 1

       # Make directory for brick mount point
       [ $dryrun -eq 0 ] && mkdir -p $brickpath/brick$dev_count

       # Create entry in /etc/fstab
       create_fstab_entry $vgname $lvname $brickpath/brick$dev_count || exit
1

       # Mount all the bricks.
```

```
       [ $dryrun -eq 0 ] && mount -a

       (( count++ ))
       (( dev_count++ ))
            echo
    done
}

function tune_performance {

    echo "---- Performance tune for $workload storage ----"
    tuned-adm profile $tune_profile
}


function channels_error {
   declare -a reg_channels=(`rhn-channel --list`)
   echo "ERROR: All required channels are not registered!"
   echo -e "Required Channels:\n\trhel-x86_64-server-6.4.z\n\trhel-x86_64-
server-sfs-6.4.z\n\trhel-x86_64-server-6-rhs-2.1"
   echo -e "Registered Channels:"
   for chan in "${reg_channels[@]}"
   do
         echo -e "\t$chan"
   done
   return 1
}


function check_channels {

   declare -a reg_channels=(`rhn-channel --list`)
   if [ ${#reg_channels[@]} -lt 3 ]
   then
      channels_error
      return 1
   fi

   correct=0
   for chan in "${reg_channels[@]}"
   do
      if [ "$chan" == "rhel-x86_64-server-6.4.z" \
           -o "$chan" == "rhel-x86_64-server-sfs-6.4.z" \
           -o "$chan" == "rhel-x86_64-server-6-rhs-2.1" \
         ]
      then
         (( correct++ ))
      fi
   done

   if [ $correct -ne 3 ]
   then
      channels_error
      return 1
   fi
```

```
   echo -e "Registered Channels:"
   for chan in "${reg_channels[@]}"
   do
         echo -e "\t$chan"
   done
   return 0
}


function rhn_register_update {

    profile_name=`hostname -s`
    profile_name=RHS_$profile_name
    rhn_register

    echo "---- Register Channels ----"
    read -p "RHN Login: " rhn_login
    read -s -p "RHN Password: " rhn_password
    echo ""
    rhn-channel --verbose --user $rhn_login --password $rhn_password \
        --add --channel=rhel-x86_64-server-sfs-6.4.z
    rhn-channel --verbose --user $rhn_login --password $rhn_password \
        --add --channel=rhel-x86_64-server-6-rhs-2.1

    check_channels || return 1
    echo "System registered to the correct Red Hat Channels!"
    if yesno "Do you want to apply updates now? "
    then
        echo "---- Apply Updates ----"
        yum -y update
    fi
}


function main {

    while getopts ":rnhu:" OPT; do
    case "$OPT" in
        u)
            case $OPTARG in
                object)
                    workload=$OPTARG
                    inode_size=256
                    ;;
                virtual)
                    workload=$OPTARG
                    tune_profile=rhs-virtualization
                    ;;
                *)
                    echo "Unrecognized option."
                    usage # print usage and exit
            esac
            ;;
       n)
```

```
            dryrun=1
            ;;
        r)
            skip_registration=1
            ;;
        h)
            usage # print usage and exit
            ;;
        \?)
            echo "Invalid option: -$OPTARG"
            usage # print usage and exit
            ;;
        :)
            echo "Option -$OPTARG requires an argument."
            usage # print usage and exit
            ;;
    esac
    done
    echo "Setting workload to $workload."

    # Check whether it is a physical or a virtual deployment
    tempvar=(`lvmdiskscan | grep /dev/sda`)
    ret=$?
    if [ $ret -eq 0 ]
    then
        echo "Physical deployment!"
        disk_path=/dev/sd
        root_disk_path=/dev/sda
    else
         tempvar=(`lvmdiskscan | grep /dev/vda`)
         ret=$?
         if [ $ret -eq 0 ]
         then
             echo "Virtual deployment!"
             disk_path=/dev/vd
             root_disk_path=/dev/vda
         fi
    fi

    if [ "$disk_path" == "NODISK" ]
    then
        echo "Unknown Deployment : Could not find physical (/dev/sda) or
virtual (/dev/vda) devices!"
        echo "exiting ..."
        return 1
    fi

    # Brick creation
    create_bricks

    # If dry run then exit
    [ $dryrun -eq 1 ] && return 0

    # Invoke tuned profile
    tune_performance
```

```
    # Register and update with RHN
    if [ $skip_registration -ne 1 ]
    then
        rhn_register_update
    fi

}

# Call Main
main "$@";
```

Appendix H: Configuration Files Copy

A copy of the relevant configuration files and RHS Server Initialization script can be downloaded at the following link:

https://access.redhat.com/site/node/410303/40/1

Appendix I: Revision History

Revision 2.0	September 2013	Balaji Jayavelu
Revision 1.0	June 2013	Balaji Jayavelu
Initial Release