STORAGE MANAGEMENT USING OPENFILER – II

The first part of this series gave readers step-by-step instructions to build Openfiler from scratch. This second part covers two additional important features of Openfiler: bonded interfaces and software RAID. We also have a set of crucial troubleshooting instructions, if the bonded interface configuration fails during setup.

The earlier article used a standard PC based on an Intel dual-core 2.8 GHz CPU, 1 GB RAM, a single 100 MBps Ethernet card, and a SATA 80 GB hard disk. To create bonded interface and software RAID configurations, one Fast Ethernet card and two 500 GB hard disks were added to this original hardware.

Before starting the actual set-up, let us understand the basics of bonded interfaces and various RAID levels.

Bonded interface: A standard SATA II hard disk gives a write speed of 6 Gigabits per second. Fast Ethernet transfers 100 Megabits per second and Gigabit Ethernet 1 Gigabit per second. Thus, the network speed is six to sixty times lower than hard disk write speeds. Bonding two interfaces into a single one bridges this speed difference. Here, typically, two

independent Ethernet interfaces with their own IP addresses are bonded to be accessed as a single interface by an additional third IP address, thus creating twice the bandwidth. Bonding is also possible for more than two interfaces. Bonded interface also provide redundancy if one of the Ethernet cards interface fail.

RAID: Openfiler supports RAID levels 0, 1, 5, 6 and 10, which are explained in brief for ready reference. For more information, please see the Wikipedia article on RAID.

1. **RAID 0 (block-level striping without parity):** Given two hard disks (the minimum), a block of data to be written is split into two equal parts and written on the two disks. This almost doubles write performance. There is no redundancy; if one disk fails, all data is lost.
2. **RAID 1 (mirroring without parity or striping):** This requires a minimum of two hard disks. Whatever is written on Disk A is duplicated on Disk B, providing 1-to-1 redundancy. This level can sustain failure of one disk without losing data. When a faulty disk is replaced, the RAID controller automatically synchronises the data.
3. **RAID 5 (block-level striping with distributed parity):** This level requires a minimum of three hard disks and tolerates failure of one disk without loss of data. Given three disks, A, B and C, and three data blocks to be written on them, Table 1 shows how the blocks are stored, after splitting each into two equal parts.

Table 1: RAID 5 storage structure

Block	Data	Parity
First block	Disk A and B	Disk C
Second block	Disk B and C	Disk A
Third block	Disk C and A	Disk B

4. The parity information is calculated using the logical Exclusive-OR (XOR) function. Table 2 helps understand the function, and how it is used to reconstruct data to be written on a replacement hard disk.

Table 2: XOR truth table and disks where data and parity are written

Data A	Data B	Parity

Table 2: XOR truth table and disks where data and parity are written		
Data A	**Data B**	**Parity**
0	0	0
0	1	1
1	0	1
1	1	0
Disk A	Disk B	Disk C

5. With XOR, when data A and data B are equal, the parity is 0; when not equal, parity is 1. Now let's assume Disk A fails. At this point, we have correct values for data B and parity, stored on disks B and C. After the faulty disk is replaced, the RAID controller recalculates and writes the correct data to it, using the following rules:

   - If Data B and Parity both are 0, Data A is 0.
   - If Data B is 1 and Parity is 0, Data A is 1.
   - If Data B is 0 and Parity is 1, Data A is 1.
   - If Data B and Parity both are 1, Data A is 0.

   There are two disadvantages of using this RAID level:

   - During synchronisation (rebuilding the RAID array), the array's read-write performance is greatly reduced due to calculation overhead.
   - If any other disk fails during synchronisation, the whole array is destroyed.
6. **RAID 6 (block-level striping with double distributed parity):** This level uses two parity disks instead of one as in the case of RAID 5; thus, it can tolerate failure of two hard disks in an array, at a time.
7. **RAID 10 (stripe of mirrors):** Combines RAID 0 and 1. Two RAID 1 arrays are striped using RAID 0, thus providing speed as well as redundancy.

Now, after the basics, let's proceed with Openfiler configuration of bonded interfaces and RAID levels.

Bonded interface configuration

First, the prerequisites: a minimum of two Ethernet cards (eth0 and eth1) and two additional IP addresses in the same range as the Openfiler installation (the third IP is for the bonded interface bond0).

Now that we have the bare minimum, open the Openfiler Web interface (`https://ipaddress:446`), log in and go to *System –> Network Interface Configuration*. Edit the eth1 configuration, and enter the IP address and subnet mask for this interface (see Figure 1).

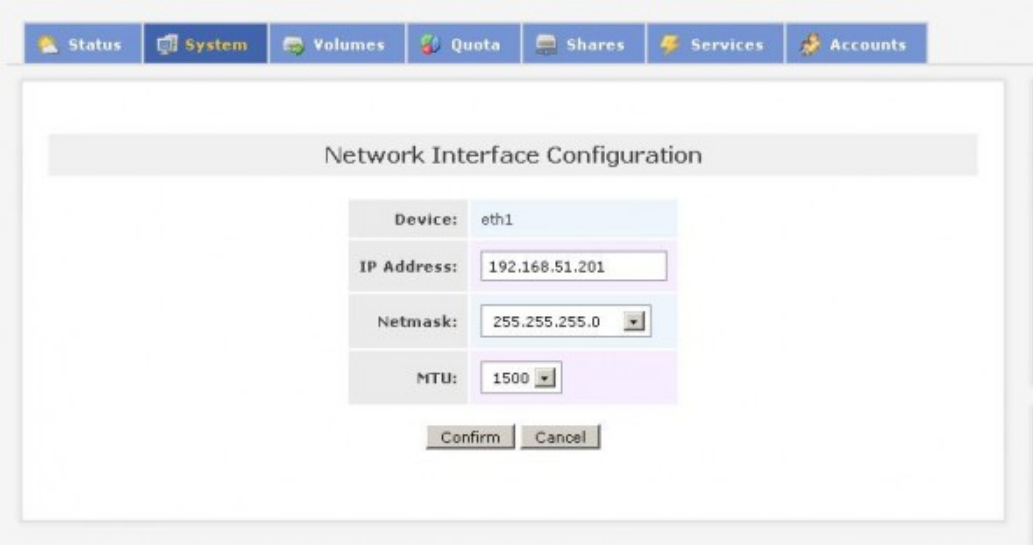

Figure 1: Network interface configuration

Proceed with bonding the two interfaces once the second interface is configured (Figures 2 and 3).

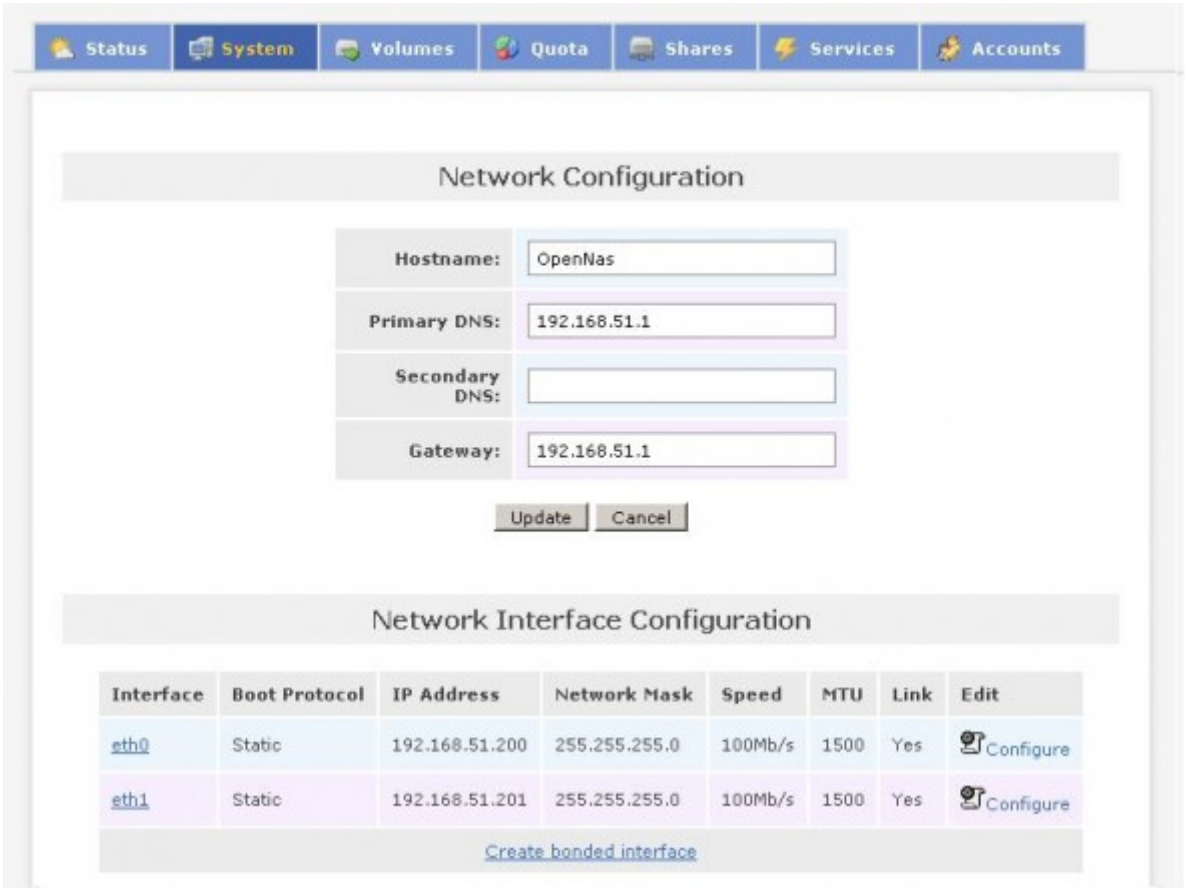

Figure 2: Bonding two interfaces

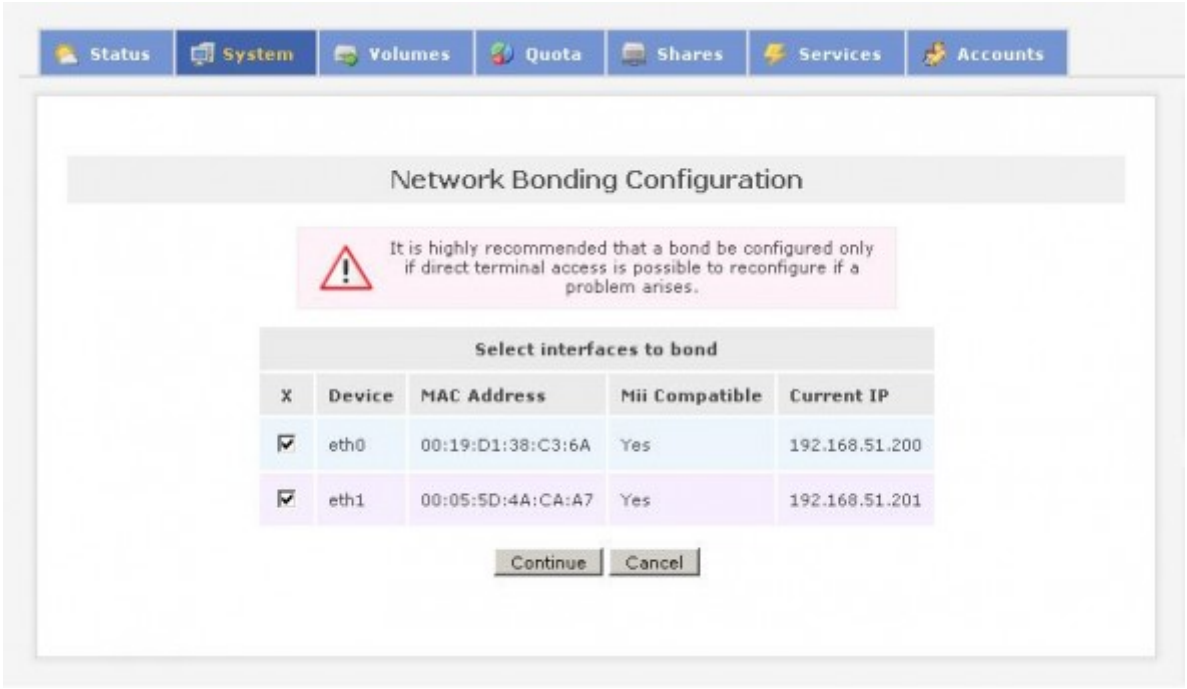

Figure 3: Network bonding configuration1

Continue to give the bonded IP address and subnet mask, leaving all other parameters at their default values (Figure 4).

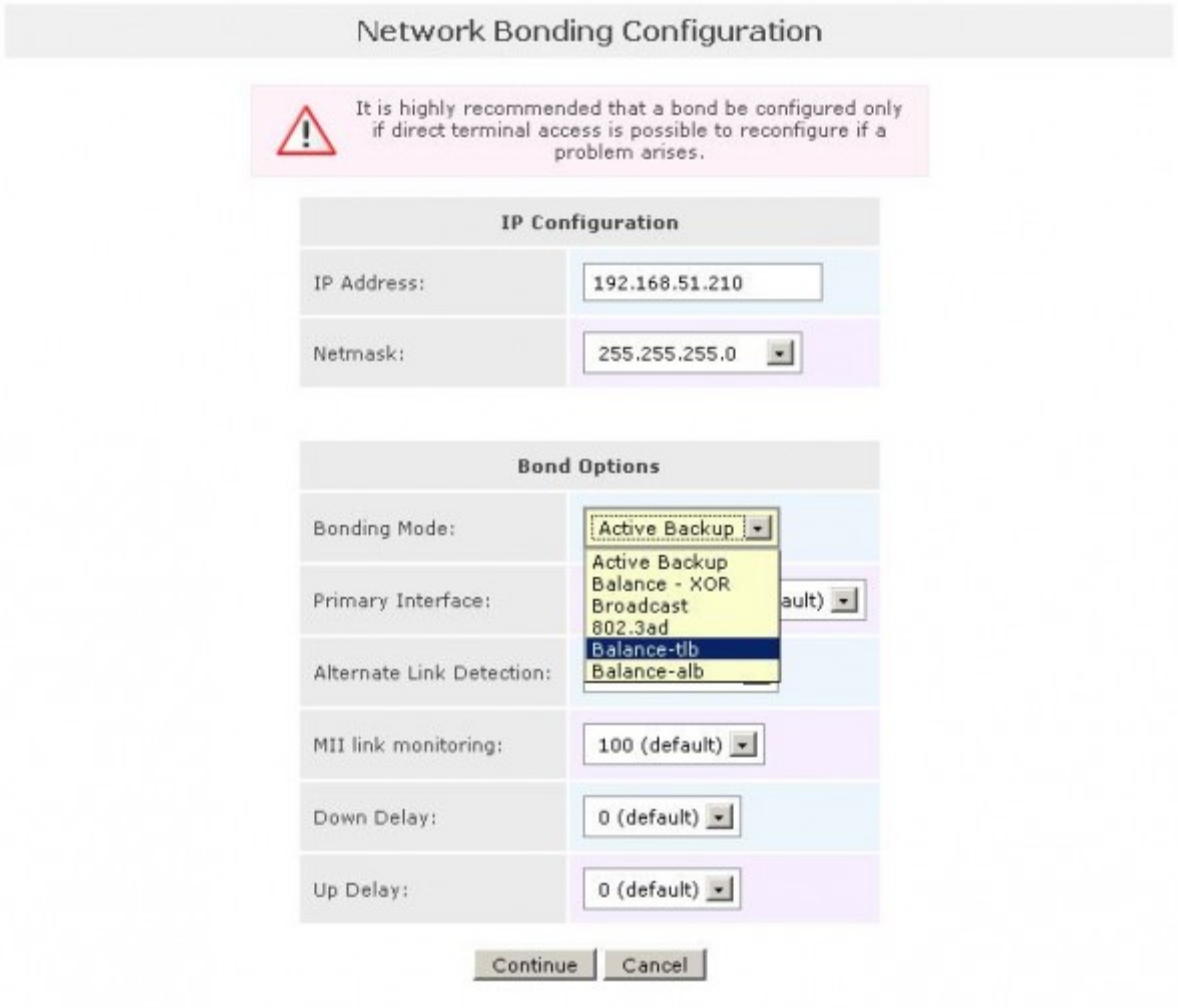

Figure 4: Network bonding configuration2

Select the bonding type as balance-alb (adaptive load balancing). Various other modes available provide combinations of load balancing methods and failover. Verify the new network configuration (Figure 5).

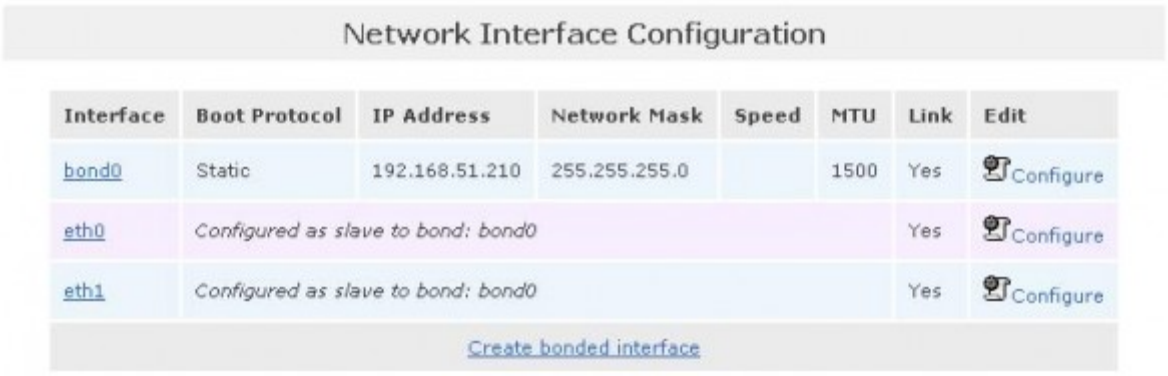

Figure 5: Configured bonded interfaces

Now, you won't be able to access the Web GUI via the eth0 address. Instead, connect to`https://bond0address:446`.

Caution: Do not abandon the configuration in between. It is highly recommended that a bond be configured only if direct terminal access is possible to reconfigure, in case a problem arises.

RAID configuration

Now, let us proceed to add RAID volumes to the box. We shall create RAID 1 100 GB volumes on the two new 500 GB hard disks.

Proceed to *Volume –> Block devices*. This (Figure 6) reflects all hard disks. Openfiler detects SATA hard disks as SCSI, but check the description — it is still ATA with exact model numbers. You can easily figure out from the description that the sda and sdc hard disks are Seagate, and sdb is Hitachi.

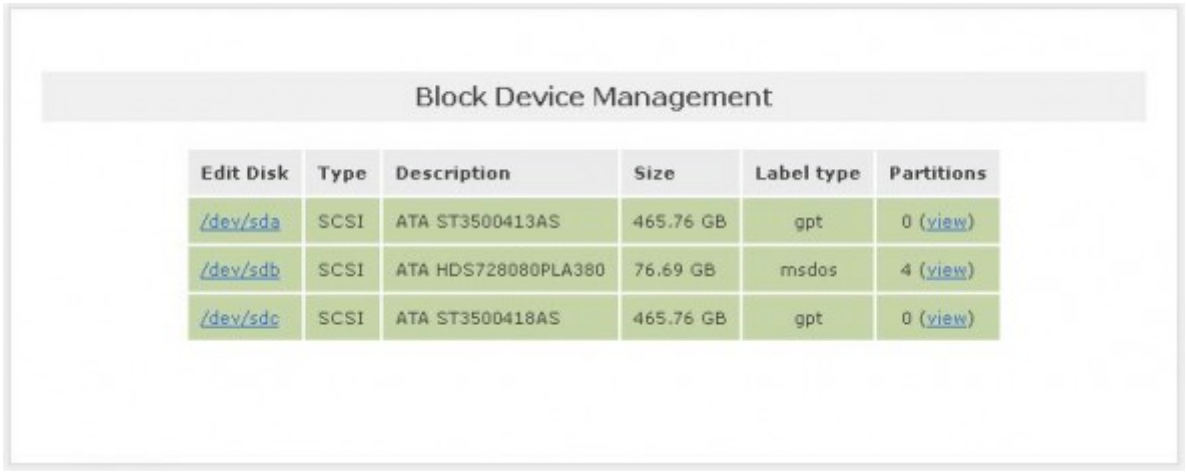

Figure 6: Block device management

Define 100 GB RAID array partitions on `/dev/sda` and `/dev/sdc` (Figure 7).

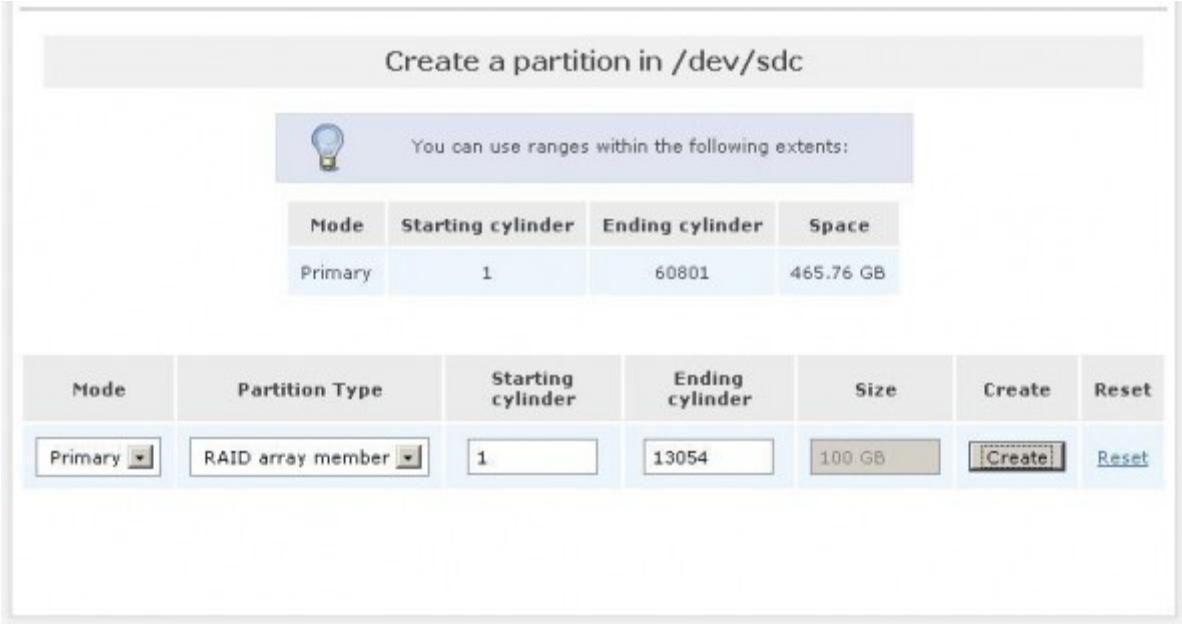

Figure 7: Creating RAID partition

Select *Volume -–&ft; Software RAID* and create a new RAID array `/dev/md0` (Figure 8).

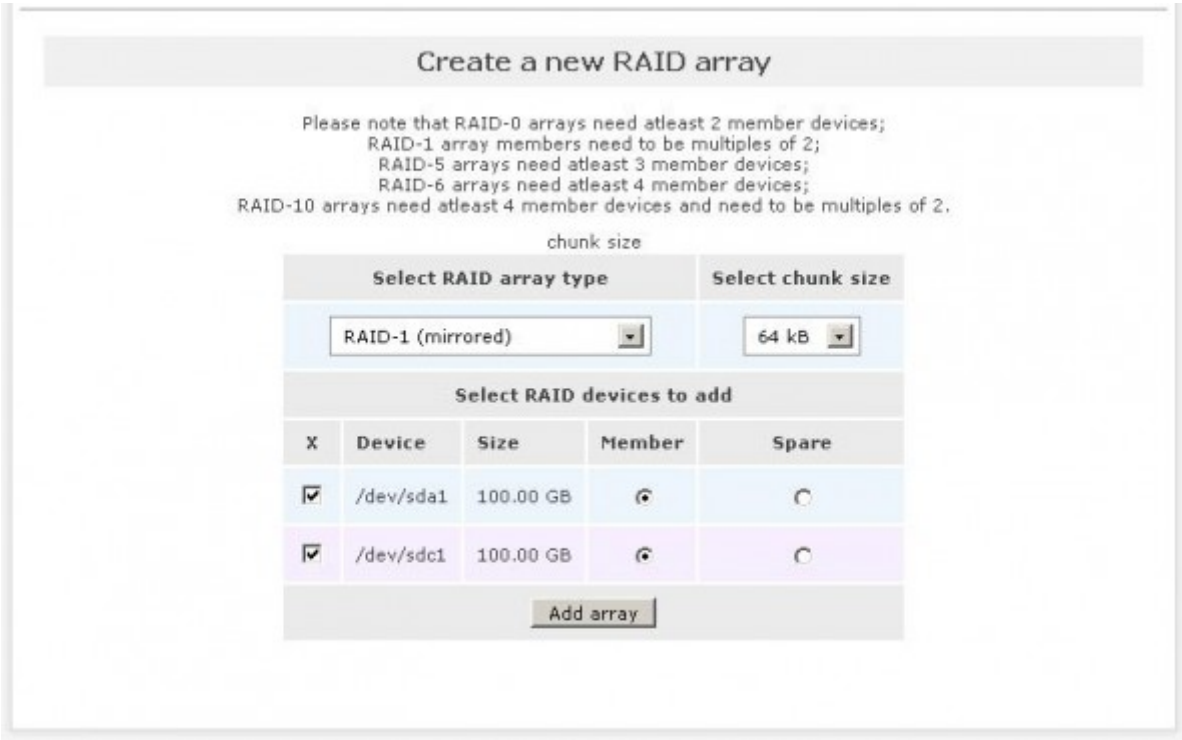

Figure 8: Creating a new RAID array

Create a new volume group (let's name it "raid1group" in `/dev/md0` (Figures 9 and 10).

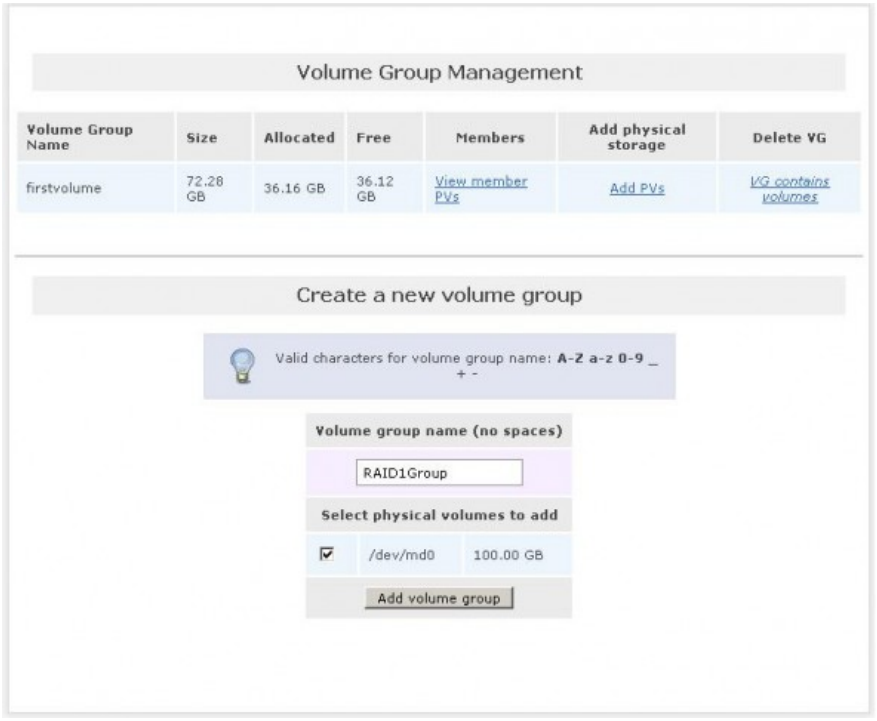

Figure 9: Creating new volume group in RAID array

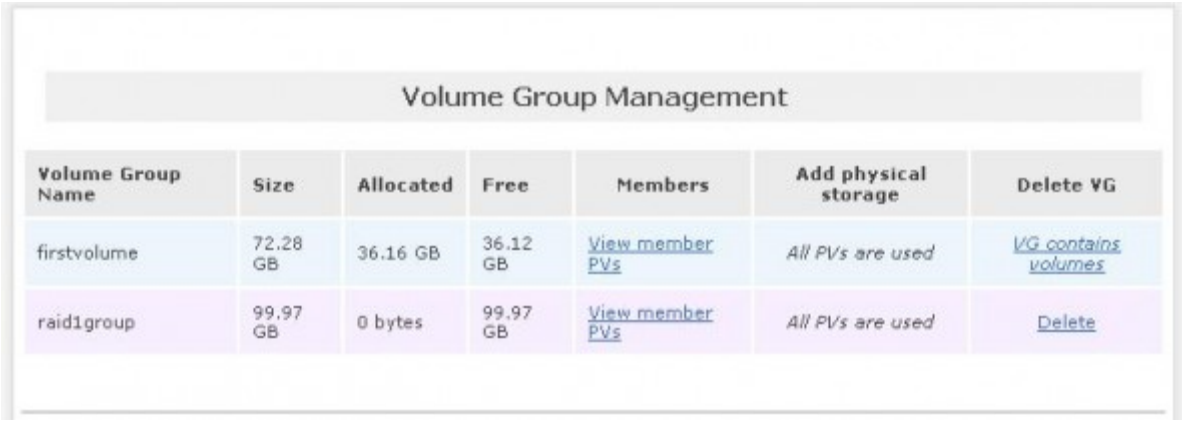

Figure 10: raid1group array successfully created

The first article explained addition of volumes for firstvolume. Exactly the same steps are required to be followed to create volumes on raid1group. (For installation purposes, you can treat a RAID volume as any standard volume created in Openfiler.) So, proceed with: *Add Volume –> Raid1Group*. Give the volume name as "Database", and description as "database storage". Allot the full 100 GB space to the volume, and select ext3 partition, to finally create the volume. The time required for creating the volume will depend on machine speed.

Proceed to define groups, users, and assign quotas to groups and users. Please refer to theprevious article for detailed steps.

Note: Once configuration is complete, be sure to save a backup of the system configuration from the System menu, and LDAP configuration from the *LDAP Setup* menu.

The benefits of using a separate disk for OS, and two additional disks for RAID 1 volumes, are:

1. The operating system resides on a separate hard disk, and doesn't occupy RAID space.
2. Important data is mirrored in a RAID 1 volume.
3. If the OS disk fails, you can install a fresh OS and restore the backup of the configuration, to restore the Openfiler configuration.
4. If one of the 500 GB hard disks fails, you can replace that hard disk and reconstruct the RAID 1 array.

Important troubleshooting tips for bonded interface

If you misconfigure or cancel the bonded interface setup, the GUI access to Openfiler stops working. Here is what you should do in such a case. Connect a keyboard and monitor to the Openfiler system, and log in as root. Run the following commands:

```
cd /etc/sysconfig/network-scripts ## Change directory to network
rm ifcfg-bond0 ## Remove bond0 interface
```

Ensure that `ifcfg-eth0` file has the following lines (adjusted for your configuration) or add them:

```
DEVICE=eth0
BOOTPROT=static
BROADCAST=192.168.51.255
IPADDR=192.168.51.200
NETMASK=255.255.255.0
NETWORK=192.168.51.1
ONBOOT=yes
TYPE=Ethernet
```

`GATEWAY=192.168.51.1`

Restart the network service (`/etc/init.d/network restart` or `service network restart`). Access the Web GUI at the configured eth0 IP address. Redo the bonding configuration properly.

That’s all for this time, folks!

Source : http://www.opensourceforu.com/2011/09/storage-management-using-openfiler-part-2/