

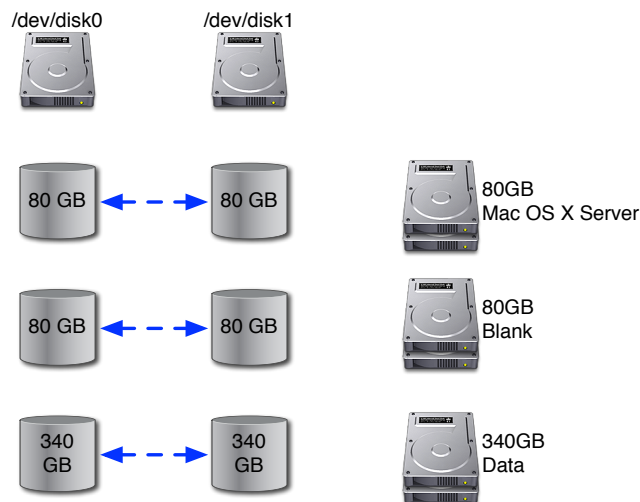
# Introduction

The Mac mini server is a sweet little machine, perfect both size-wise and cost-wise for a wide variety of purposes. The most common uses I have encountered are as an all-in-one server for a small office or home, with a direct-attached FireWire 800 external RAID as a server for a somewhat larger office, or as an Open Directory replica.

However, it comes out of the box with two 500 GB hard disk drives, each with a single partition. One of the drives has Snow Leopard Server installed on it, the other is blank.



This is not how you should configure your server's disks, as it has no redundancy and user data and system data will be on the same volume. It would be a lot better if there were two separate boot volumes (one for regular use, the second for upgrades or disaster recovery) and a data volume, and each of the volumes was redundant with RAID 1.



# The Problem

OK, so how can we pull this off? There are two ways that this can come about — RAID the disks together, then partition them, or partition them first, then RAID the partitions. It turns out that Disk Utility's RAID-1 works on partitions, not on disks. If you are putting two entire disks into a RAID-1, you are actually creating a single partition on each disk and making a RAID-1 out of the partitions.

So, we can open Disk Utility and partition each disk into three partitions, 80 GB, 80 GB, and 340 GB. So far, so good. Select the RAID tab, drag the first pair of partitions into the pane, and create the RAID-1. Repeat for the second set -- and it fails. Uh, oh, what's going on?

It turns out that there are two problems: Disk Utility won't create a second RAID from the partitions on disks where there is already a RAID set, and Disk Utility wants to put a small (134.2 MB) Boot OSX partition after each RAID partition. If you don't leave space for the Boot OSX partition, creating the RAID will fail.

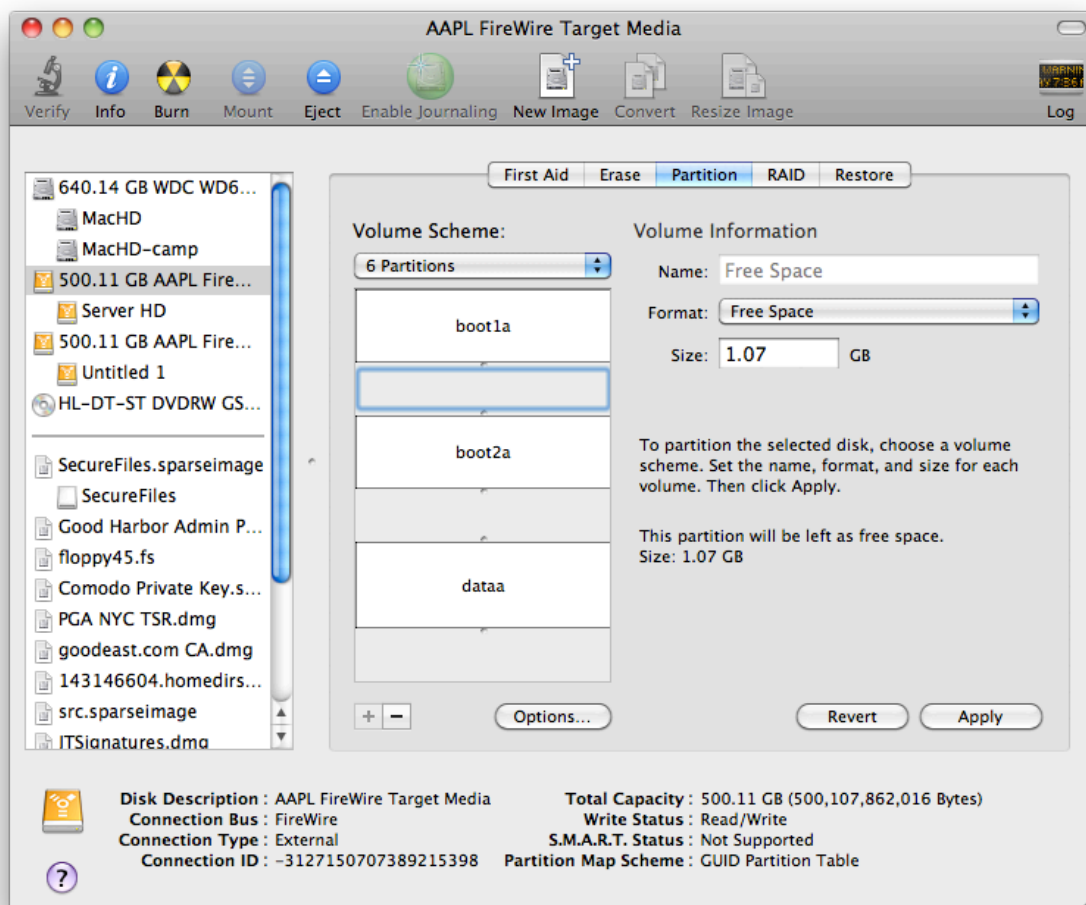
## The Solution

So what can we do? We'll handle this in two parts. First, create the partitions in Disk Utility with a sliver of free space after each of them to hold the Boot OS X partition. Then, we'll use the `diskutil` command to create the RAID sets. This procedure can be done booting from the Snow Leopard Server Install DVD, or by booting the mini into target disk mode and attaching it to another Mac. In this example the mini is booted into target disk mode and attached to another Mac, but the steps are identical when doing it booted from an Install DVD.

### Partitioning

It turns out that the smallest partition that Disk Utility will let you create is 1.07 GB, so select the first physical drive, make the volume scheme 6 partitions, and set them to:

Volume Name	Size	Format
boot1a	80 GB	Mac OS Extended (Journaled)
	1.07 GB	Free Space
boot2a	80 GB	Mac OS Extended (Journaled)
	1.07 GB	Free Space
dataa	336.9 GB	Mac OS Extended (Journaled)
	1.07 GB	Free Space



The easiest way to do this is to configure them in the following order:

- boot1a (80 GB)
- free space following boot1a (1 GB)
- boot2a (80 GB)
- free space following boot2a (1 GB)
- free space following dataa (1 GB)
- dataa (remainder of the available space)

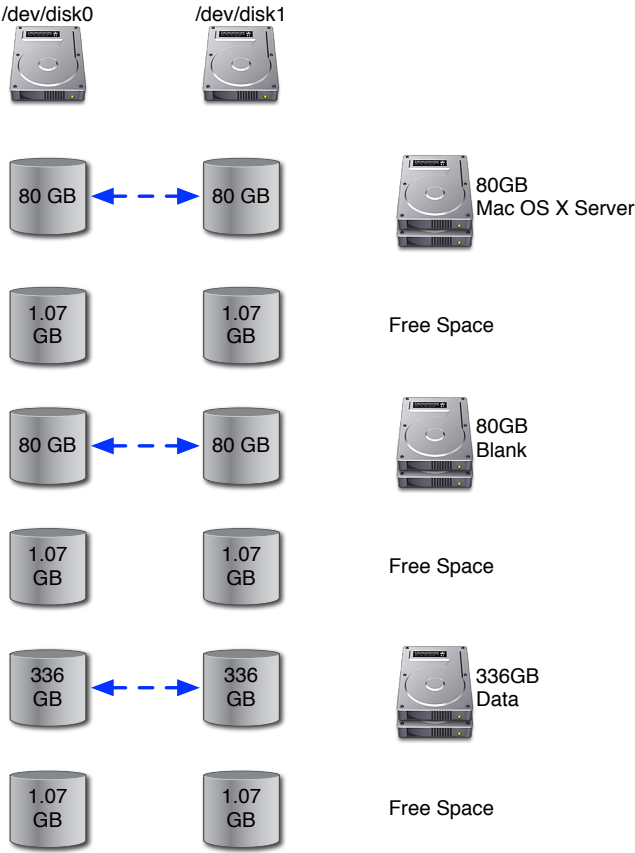
This will allow the data volume to be as large as possible without worrying about getting the size exactly right.

Repeat this with the second physical drive, but use the names and sizes:

Volume Name	Size	Format
boot1b	80 GB	Mac OS Extended (Journaled)

Volume Name	Size	Format
	1.07 GB	Free Space
boot2b	80 GB	Mac OS Extended (Journaled)
	1.07 GB	Free Space
datab	336.9 GB	Mac OS Extended (Journaled)
	1.07 GB	Free Space

The final setup should look like:



### Creating the RAID Sets

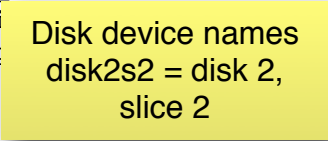
To create the RAID sets, we need to go to Terminal.app and use the `/usr/sbin/diskutil` command. First, list out all of the disks using `diskutil list` — we need the device numbers of the partitions.

```
Paul-Suh-MBP-13:~ plsuh$ diskutil list
```

```

/dev/disk0
#:                TYPE NAME                                SIZE
IDENTIFIER
0:      GUID_partition_scheme                            *640.1 GB   disk0
1:      EFI                                              209.7 MB   disk0s1
2:      Apple_HFS MachHD                                532.8 GB   disk0s2
3:      Apple_HFS MachHD-camp                          106.9 GB   disk0s3
/dev/disk1
#:                TYPE NAME                                SIZE
IDENTIFIER
0:      Apple_partition_scheme                          *524.3 MB   disk1
1:      Apple_partition_scheme                          32.3 KB     disk1s1
2:      Apple_partition_scheme                          524.2 MB   disk1s2
/dev/disk2
#:                TYPE NAME                                SIZE
IDENTIFIER
0:      GUID_partition_scheme                            *500.1 GB   disk2
1:      EFI                                              209.7 MB   disk2s1
2:      Apple_HFS boot1a                                80.0 GB     disk2s2
3:      Apple_HFS boot2a                                80.0 GB     disk2s3
4:      Apple_HFS dataa                                  336.9 GB   disk2s4
/dev/disk3
#:                TYPE NAME                                SIZE
IDENTIFIER
0:      GUID_partition_scheme                            *500.1 GB   disk3
1:      EFI                                              209.7 MB   disk3s1
2:      Apple_HFS boot1b                                80.0 GB     disk3s2
3:      Apple_HFS boot2b                                80.0 GB     disk3s3
4:      Apple_HFS datab                                  336.9 GB   disk3s4

```



The partitions highlighted in red are the ones we will use for creating the RAID sets. Notice that the free space partitions don't show up — they're just free space, not real partitions. (The device names are only for the example here; they may change depending on your actual setup. Check the output of `diskutil` for yourself on your system — **do not depend on the partitions having the same device numbers as are listed here!**)

The `diskutil` command that will create the first RAID set will be applied to the `disk2s2` and `disk3s2` partitions in this example. Adjust the command to fit the device names that come from your list of devices. The command and its output are:

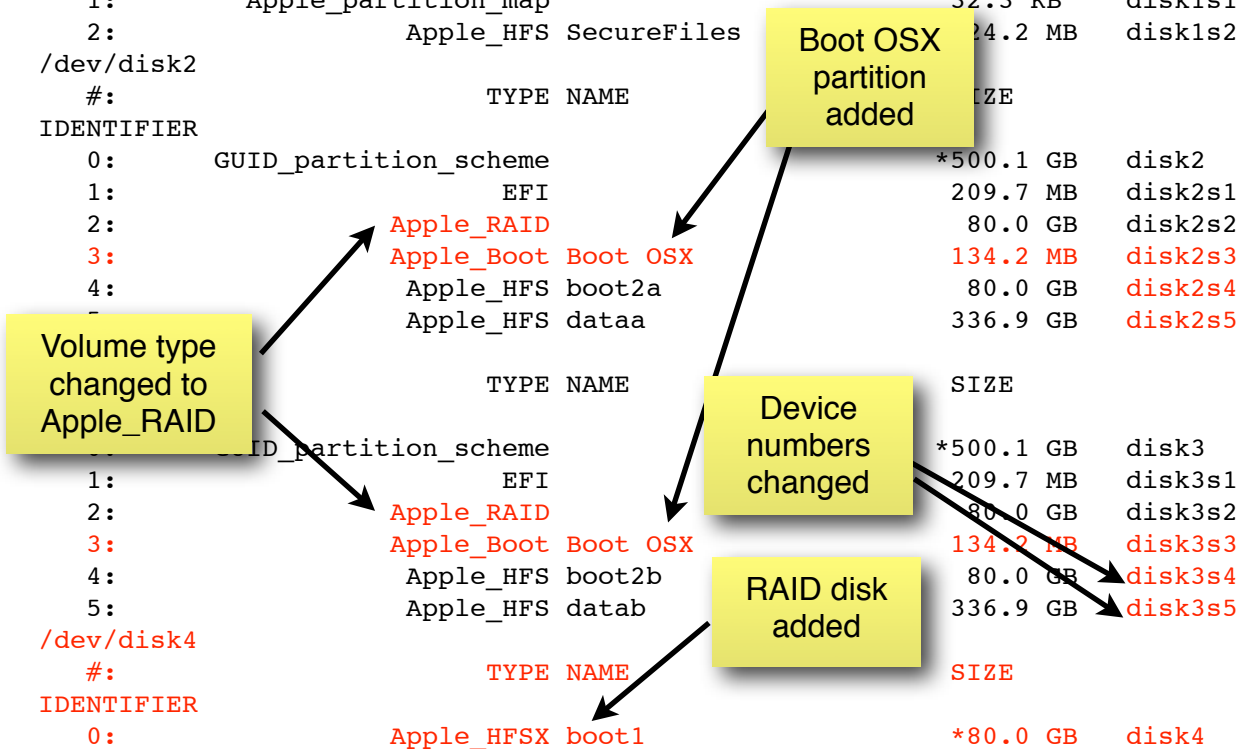
```

Paul-Suh-MBP-13:~ plsu$ diskutil AppleRAID create mirror boot1 jhfsx \
    disk2s2 disk3s2
Started RAID operation
$<3>Unmounting proposed new member disk2s2
$<3>Unmounting proposed new member disk3s2
$<3>Adding booter for RAID partition disk2s2
$<3>Adding booter for RAID partition disk3s2
$<3>Creating RAID set
$<3>Bringing RAID partitions online
$<3>Waiting for new RAID to spin up "9D9A1B0D-C16A-46A2-B2A3-D36F4D0D5310"
$<3>Finished RAID operation

```

Once this is done, run `diskutil list` again to see what has changed.

```
Paul-Suh-MBP-13:~ plsuh$ diskutil list
/dev/disk0
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      GUID_partition_scheme          *640.1 GB    disk0
1:      EFI                            209.7 MB    disk0s1
2:      Apple_HFS MacHD                 532.8 GB    disk0s2
3:      Apple_HFS MacHD-camp            106.9 GB    disk0s3
/dev/disk1
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      Apple_partition_scheme          *524.3 MB    disk1
1:      Apple_partition_map             32.3 KB     disk1s1
2:      Apple_HFS SecureFiles           24.2 MB     disk1s2
/dev/disk2
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      GUID_partition_scheme          *500.1 GB    disk2
1:      EFI                            209.7 MB    disk2s1
2:      Apple_RAID                      80.0 GB     disk2s2
3:      Apple_Boot Boot OSX             134.2 MB    disk2s3
4:      Apple_HFS boot2a                80.0 GB     disk2s4
5:      Apple_HFS dataa                 336.9 GB    disk2s5
/dev/disk3
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      GUID_partition_scheme          *500.1 GB    disk3
1:      EFI                            209.7 MB    disk3s1
2:      Apple_RAID                      80.0 GB     disk3s2
3:      Apple_Boot Boot OSX             134.2 MB    disk3s3
4:      Apple_HFS boot2b                80.0 GB     disk3s4
5:      Apple_HFS datab                 336.9 GB    disk3s5
/dev/disk4
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      Apple_HFSX boot1                *80.0 GB    disk4
```



I have highlighted the changes that occurred. Note the added Boot OSX partitions, added RAID device (`disk4`), and the fact that **the device numbers have changed for the partitions on the physical disks.** (`boot2a` was `disk2s3`, but is now `disk2s4`.)

We can now repeat the RAID creation process for the second boot partition, using the updated device numbers:

```
Paul-Suh-MBP-13:~ plsuh$ diskutil AppleRAID create mirror boot2 jhfsx \
    disk2s4 disk3s4
Started RAID operation
$<3>Unmounting proposed new member disk2s4
$<3>Unmounting proposed new member disk3s4
$<3>Adding booter for RAID partition disk2s4
$<3>Adding booter for RAID partition disk3s4
```

```

$<3>Creating RAID set
$<3>Bringing RAID partitions online
$<3>Waiting for new RAID to spin up "CCA0F892-B6C9-461A-A3F2-E3B805F62F50"
$<3>Finished RAID operation

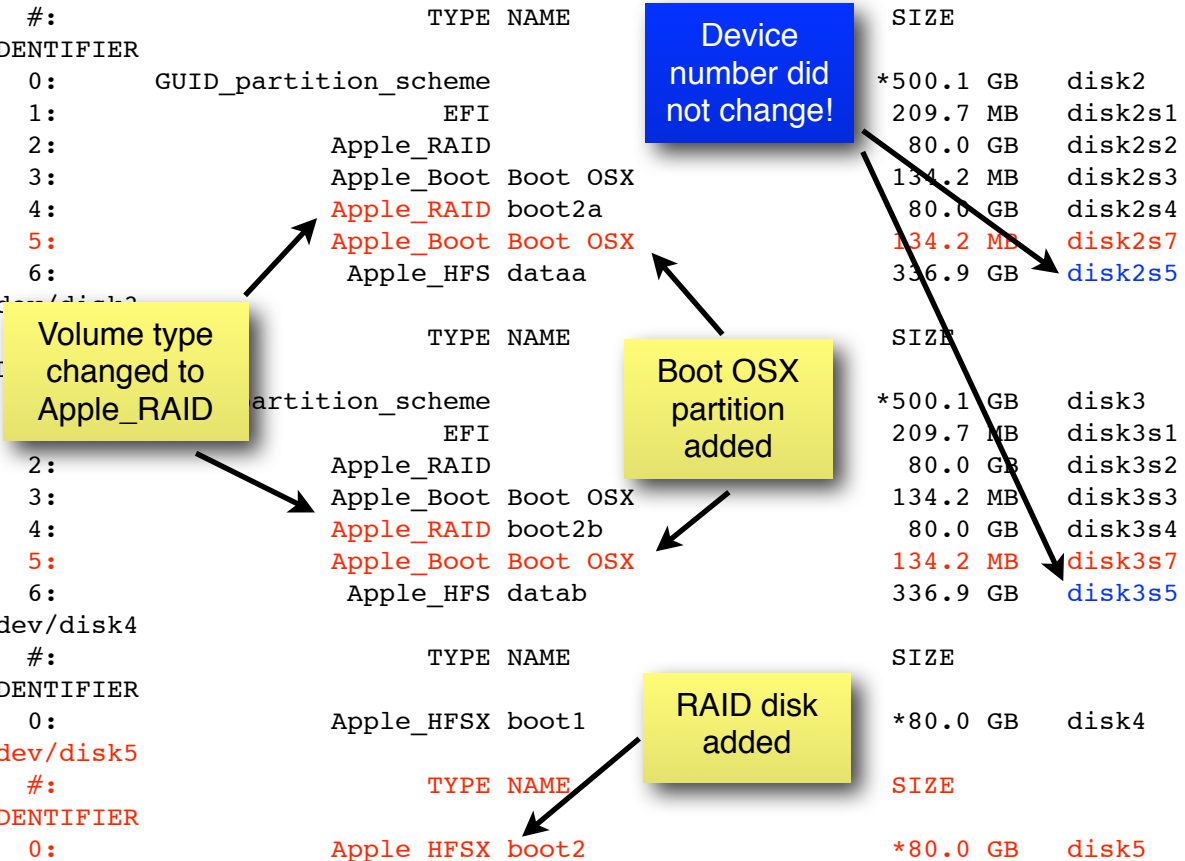
```

Once this is done, run `diskutil list` yet again to see what has changed.

```

Paul-Suh-MBP-13:~ plsuh$ diskutil list
/dev/disk0
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      GUID_partition_scheme            *640.1 GB    disk0
1:      EFI                               209.7 MB    disk0s1
2:      Apple_HFS MachHD                  532.8 GB    disk0s2
3:      Apple_HFS MachHD-camp            106.9 GB    disk0s3
/dev/disk1
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      Apple_partition_scheme           *524.3 MB    disk1
1:      Apple_partition_map              32.3 KB     disk1s1
2:      Apple_HFS SecureFiles            524.2 MB    disk1s2
/dev/disk2
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      GUID_partition_scheme            *500.1 GB    disk2
1:      EFI                               209.7 MB    disk2s1
2:      Apple_RAID                        80.0 GB     disk2s2
3:      Apple_Boot Boot OSX                134.2 MB    disk2s3
4:      Apple_RAID boot2a                  80.0 GB     disk2s4
5:      Apple_Boot Boot OSX                134.2 MB    disk2s7
6:      Apple_HFS dataa                   336.9 GB    disk2s5
/dev/disk3
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      GUID_partition_scheme            *500.1 GB    disk3
1:      EFI                               209.7 MB    disk3s1
2:      Apple_RAID                        80.0 GB     disk3s2
3:      Apple_Boot Boot OSX                134.2 MB    disk3s3
4:      Apple_RAID boot2b                  80.0 GB     disk3s4
5:      Apple_Boot Boot OSX                134.2 MB    disk3s7
6:      Apple_HFS datab                   336.9 GB    disk3s5
/dev/disk4
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      Apple_HFSX boot1                   *80.0 GB    disk4
/dev/disk5
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:      Apple_HFSX boot2                   *80.0 GB    disk5

```



It's very similar to the results of the first RAID set creation, but notice that the device numbers of the `dataa` and `datab` partitions did **not** change this time. *This is why it is critical to check the device numbers for the partitions after each RAID set creation step.*

Now we can create the last RAID set.

```
Paul-Suh-MBP-13:~ plsuh$ diskutil AppleRAID create mirror data jhfsx \  
disk2s5 disk3s5  
Started RAID operation  
$<3>Unmounting proposed new member disk2s5  
$<3>Unmounting proposed new member disk3s5  
$<3>Adding booter for RAID partition disk2s5  
$<3>Adding booter for RAID partition disk3s5  
$<3>Creating RAID set  
$<3>Bringing RAID partitions online  
$<3>Waiting for new RAID to spin up "C3B780CC-2397-4251-9EBB-20F2129239FF"  
$<3>Finished RAID operation
```

diskutil list will show you the final result:

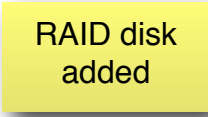
```
Paul-Suh-MBP-13:~ plsuh$ diskutil list  
/dev/disk0  
#:                TYPE NAME                SIZE  
IDENTIFIER  
0:      GUID_partition_scheme                *640.1 GB    disk0  
1:                EFI                        209.7 MB     disk0s1  
2:      Apple_HFS MachHD                    532.8 GB     disk0s2  
3:      Apple_HFS MachHD-camp                106.9 GB     disk0s3  
/dev/disk1  
#:                TYPE NAME                SIZE  
IDENTIFIER  
0:      Apple_partition_scheme                *524.3 MB     disk1  
1:      Apple_partition_map                    32.3 KB       disk1s1  
2:      Apple_HFS SecureFiles                 524.2 MB     disk1s2  
/dev/disk2  
#:                TYPE NAME                SIZE  
IDENTIFIER  
0:      GUID_partition_scheme                *500.1 GB     disk2  
1:                EFI                        209.7 MB     disk2s1  
2:      Apple_RAID                          80.0 GB       disk2s2  
3:      Apple_Boot Boot OSX                    134.2 MB     disk2s3  
4:      Apple_RAID boot2a                       80.0 GB       disk2s4  
5:      Apple_Boot Boot OSX                    134.2 MB     disk2s7  
6:      Apple_RAID dataaa                       336.9 GB     disk2s5  
7:      Apple_Boot Boot OSX                    134.2 MB     disk2s8  
/dev/disk3  
#:                TYPE NAME                SIZE  
IDENTIFIER  
0:      GUID_partition_scheme                *500.1 GB     disk3  
1:                EFI                        209.7 MB     disk3s1  
2:      Apple_RAID                          80.0 GB       disk3s2  
3:      Apple_Boot Boot OSX                    134.2 MB     disk3s3  
4:      Apple_RAID boot2b                       80.0 GB       disk3s4  
5:      Apple_Boot Boot OSX                    134.2 MB     disk3s7  
6:      Apple_RAID datab                       336.9 GB     disk3s5  
7:      Apple_Boot Boot OSX                    134.2 MB     disk3s8  
/dev/disk4
```



```

#:                                TYPE NAME                                SIZE
IDENTIFIER
0:                                Apple_HFSX boot1                                *80.0 GB   disk4
/dev/disk5
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:                                Apple_HFSX boot2                                *80.0 GB   disk5
/dev/disk6
#:                                TYPE NAME                                SIZE
IDENTIFIER
0:                                Apple_HFSX data                                *336.9 GB  disk6

```



As a final check, you can use the `diskutil AppleRAID list` command to see the details of the RAID sets.

```
Paul-Suh-MBP-13:~ plsuh$ diskutil AppleRAID list
AppleRAID sets (3 found)
```

```

=====
Name:                                boot1
Unique ID:                           9D9A1B0D-C16A-46A2-B2A3-D36F4D0D5310
Type:                                  Mirror
Status:                                Online
Size:                                  80.0 GB (79999978700 Bytes)
Rebuild:                               automatic
Device Node:                           disk4
=====

```

```

#   Device Node      UUID                                     Status
-----
0   disk2s2          D758356D-6D43-4A79-9D63-5EE49AC74FA2    Online
1   disk3s2          7FB0F076-A55F-4169-BB4D-5BF5FC4BBC28    Online
=====

```

```

Name:                                boot2
Unique ID:                           CCA0F892-B6C9-461A-A3F2-E3B805F62F50
Type:                                  Mirror
Status:                                Online
Size:                                  80.0 GB (79999978700 Bytes)
Rebuild:                               automatic
Device Node:                           disk5
=====

```

```

#   Device Node      UUID                                     Status
-----
0   disk2s4          87F2E770-C200-4D08-BB6D-0A3855B1492F    Online
1   disk3s4          EE0C03FA-2AD2-475A-B229-AB7200A2BA9A    Online
=====

```

```

Name:                                data
Unique ID:                           C3B780CC-2397-4251-9EBB-20F2129239FF
Type:                                  Mirror
Status:                                Online
Size:                                  336.9 GB (336884831498 Bytes)
Rebuild:                               automatic
Device Node:                           disk6
=====

```

#	Device Node	UUID	Status
0	disk2s5	3C01B22B-623A-4F6A-8518-D9606ECBF06F	Online
1	disk3s5	853FB55C-D037-4497-9FDE-71BCAB977B42	Online

## Conclusion

Our Mac mini server is now ready to have Leopard Server or Snow Leopard Server installed on one of the two boot volumes. 80 GB is more than enough space for a boot volume, and we have over 300 GB available as an internal data volume. All of the volumes are RAID-1 mirrors, and the second boot volume is available for upgrades or disaster recovery.