Create a Swap File

If there is no free partition, then We will create a file called `swapfile` in our root (/) directory. The file must allocate the amount of space we want for our swap file.

`fallocate` program. This command creates a file of a preallocated size instantly, without actually having to write dummy contents.

We can create a 4 Gigabyte file by typing:

```
sudo fallocate -l 4G /swapfile
```

The prompt will be returned to you almost immediately. We can verify that the correct amount of space was reserved by typing:

```
ls -lh /swapfile
```

```
-rw-r--r-- 1 root root 4.0G Apr 28 17:19 /swapfile
```

As you can see, our file is created with the correct amount of space set aside.

Enabling the Swap File

Right now, our file is created, but our system does not know that this is supposed to be used for swap. We need to tell our system to format this file as swap and then enable it.

Before we do that though, we need to adjust the permissions on our file so that it isn't readable by anyone besides root. Allowing other users to read or write to this file would be a huge security risk. We can lock down the permissions by typing:

```
sudo chmod 600 /swapfile
```

Verify that the file has the correct permissions by typing:

```
ls -lh /swapfile
```

```
-rw------- 1 root root 4.0G Apr 28 17:19 /swapfile
```

As you can see, only the columns for the root user have the read and write flags enabled.

Now that our file is more secure, we can tell our system to set up the swap space by typing:

```
sudo mkswap /swapfile
```

```
Setting up swapspace version 1, size = 4194300 KiB
no label, UUID=e2f1e9cf-c0a9-4ed4-b8ab-714b8a7d6944
```

Our file is now ready to be used as a swap space. We can enable this by typing:

```
sudo swapon /swapfile
```

We can verify that the procedure was successful by checking whether our system reports swap space now:

```
sudo swapon -s
```

```
Filename                        Type            Size    Used    Priority
/swapfile                       file            4194300 0       -1
```

We have a new swap file here. We can use the `free` utility again to corroborate our findings:

```
free -m
```

```
             total       used       free     shared    buffers     cached
Mem:          3953        101       3851          0          5         30
-/+ buffers/cache:         66       3887
Swap:         4095          0       4095
```

Our swap has been set up successfully and our operating system will begin to use it as necessary.

Make the Swap File Permanent

We have our swap file enabled, but when we reboot, the server will not automatically enable the file. We can change that though by modifying the `fstab` file.

Edit the file with root privileges in your text editor:

```
sudo nano /etc/fstab
```

At the bottom of the file, you need to add a line that will tell the operating system to automatically use the file you created:

```
/swapfile   none    swap    sw    0   0
```

```
According to the link below, the 'sw' option indicates that the swap partition
is to be activated with 'swapon -a' command.
```

Save and close the file when you are finished.

Tweak your Swap Settings

There are a few options that you can configure that will have an impact on your system's performance when dealing with swap.

The `swappiness` parameter configures how often your system swaps data out of RAM to the swap space. This is a value between 0 and 100 that represents a percentage.

With values close to zero, the kernel will not swap data to the disk unless absolutely necessary. Remember, interactions with the swap file are "expensive" in that they take a lot longer than interactions with RAM and they can cause a significant reduction in performance. Telling the system not to rely on the swap much will generally make your system faster.

Values that are closer to 100 will try to put more data into swap in an effort to keep more RAM space free. Depending on your applications' memory profile or what you are using your server for, this might be better in some cases.

We can see the current swappiness value by typing:

```
cat /proc/sys/vm/swappiness
```

```
60
```

For a Desktop, a swappiness setting of 60 is not a bad value. For a VPS system, we'd probably want to move it closer to 0.

We can set the swappiness to a different value by using the `sysctl` command.

For instance, to set the swappiness to 10, we could type:

```
sudo sysctl vm.swappiness=10
```

```
vm.swappiness = 10
```

This setting will persist until the next reboot. We can set this value automatically at restart by adding the line to our `/etc/sysctl.conf` file:

```
sudo nano /etc/sysctl.conf
```

At the bottom, you can add:

```
vm.swappiness=10
```

Save and close the file when you are finished.