Initial Server Setup with CentOS 6

Authored by: **ASPHostServer Administrator** *[asphostserver@gmail.com]*
Saved From: http://faq.asphosthelpdesk.com/article.php?id=176

When you first begin to access your fresh new virtual private server, there are a few early steps you should take to make it more secure. Some of the first tasks can include setting up a new user, providing them with the proper privileges, and configuring SSH.

Step One—Root Login

Once you know your IP address and root password, login as the main user, root.

It is not encouraged to use root on a regular basis, and this tutorial will help you set up an alternative user to login with permanently.

```
ssh root@123.45.67.890
```

The terminal will show:

```
The authenticity of host '69.55.55.20 (69.55.55.20)' can't be established.
ECDSA key fingerprint is 79:95:46:1a:ab:37:11:8e:86:54:36:38:bb:3c:fa:c0.
Are you sure you want to continue connecting (yes/no)?
```

Go ahead and type yes, and then enter your root password.

Step Two—Change Your Password

Currently your root password is the default one that was sent to you when you registered. The first thing to do is change it to one of your choice.

```
passwd
```

CentOS is very cautious about the passwords it allows. After you type your password, you may see a BAD PASSWORD notice. You can either set a more complex password or ignore the message—CentOS will not actually stop you from creating a short or simple password, although it will advise against it.

Step Three— Create a New User

After you have logged in and changed your password, you will not need to login again to your server as root. In this step we will make a new user, with a new password, and give them all of the root capabilities.

First, create your user; you can choose any name for your user. Here I"ve suggested Demo

```
/usr/sbin/adduser demo
```

Second, create a new user password:

```
passwd demo
```

Step Four— Root Privileges

As of yet, only root has all of the administrative capabilities. We are going to give the new user the root privileges.

When you perform any root tasks with the new user, you will need to use the phrase "sudoâ€¢ before the command. This is a helpful command for 2 reasons: 1) it prevents the user from making any system-destroying mistakes 2) it stores all the commands run with sudo to the file "/var/log/secure' which can be reviewed later if needed.

Let"s go ahead and edit the sudo configuration. This can be done through the default editor, which in CentOS is called "vi"

```
/usr/sbin/visudo
```

Find the section called user privilege specification.

It will look like this:

```
# User privilege specification
root    ALL=(ALL)       ALL
```

Under the details of root's privileges, add the following line, granting all the permissions to your new user.

To began typing in vi, press "aâ€¢.

```
demo    ALL=(ALL)       ALL
```

Press`Escape`,`:`,`w`,`q`, then`Enter`to save and exit the file.

Step Five— Configure SSH (OPTIONAL)

Now it"s time to make the server more secure. **These steps are optional. They will make the server more secure by making login more difficult.**

Open the configuration file

```
sudo vi /etc/ssh/sshd_config
```

Find the following sections and change the information where applicable:

```
Port 25000
Protocol 2
PermitRootLogin no
UseDNS no
```

We"ll take these one by one.

Port: Although port 22 is the default, you can change this to any number between 1025 and 65536. In this example, I am using port 25000. Make sure you make a note of the new port number. You will need it to login in the future, and this change will make it more difficult for unauthorized people to log in.

PermitRootLogin: change this from yes to no to stop future root login. You will now only login as the new user.

Add this line to the bottom of the document, replacing demo with your username:

```
AllowUsers demo
```

Save and Exit

Step Six— Reload and Done!

Reload SSH, and it will implement the new ports and settings.

```
/etc/init.d/sshd reload
```

To test the new settings (don"t logout of root yet), open a new terminal window and login into your virtual server as your new user.

Don"t forget to include the new port number.

```
ssh -p 25000 demo@123.45.67.890
```

Your prompt should now say:

```
[demo@yourname ~]$
```