

## Dedicated Server (Linux) Quick Start Guide

This is a short primer which deals with the most common tasks people need when they first get a dedicated server. For more detailed instructions see the 'more information' section at the bottom of the document.

Section	How to...	Page
1	Access the Tagadab control panel	2
2	Find out your server's IP address	2
3	Access the Plesk Control Panel on your server	2
4	Get SSH access to your server	2
5	Add another user	3
6	Disallow root login	3
7	Change your password	3
8	Change your server's hostname	4
9	Set a DNS entry for your server	4
10	Set your server's reverse DNS entry	4
11	Start a website	4
12	Enable FTP access	5
13	Configure Apache	5
14	Install additional software	5
15	Get more information	6



## Access the Tagadab control panel

You can access the Tagadab control panel at [http://shop.tagadab.com/control\\_panel](http://shop.tagadab.com/control_panel)

## Find out your server's IP address

An IP address is a unique number which identifies your server on the Internet. Every server on the internet has at least one IP address. The IP address allocated to your server is displayed in the Tagadab control panel, in the Dedicated Servers section.

## Access the Plesk control panel on your server

If you have requested the Plesk control panel you can access it through a web browser at

**`https://<ipaddress>:8443`**

The username to login will be 'admin' (no quotes) and the password will be the same as the default root password for the server. If you have not ordered the Plesk control panel you must administer your Linux server using the command line, which this document will help you with. You can order the Plesk control panel after you have bought the server by calling our support number 0845 045 1102 (9am to 6pm Mon-Fri).

## Get SSH access to your server

By default your server is setup with one account (root). You will need your root password which is displayed on the Tagadab control panel, in the Dedicated Servers section. You will also need an SSH client. If you are using windows we recommend Putty [<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>]. On a Mac or Linux you can use the SSH command from the command line, for example:

**`ssh -l root <ipaddress>`**

where <ipaddress> is your server's IP address (see above).

The root account has full access to all parts of the system and should be protected by a strong password. Our installation system generates a strong password when your server is built, and it is recommended that you use a password of equal or greater strength. Do not use passwords based on dictionary words, your mother's maiden name, your favourite football team, or anything else that is easy to guess. Passwords should contain a mixture of upper and lower case characters, numbers and special characters. Failure to use a secure password may leave your system open to attackers who could take control of your server and its data.

## Add another user

1. get an ssh shell on your server
2. type 'adduser' followed by the username of the new account and follow the prompts.
3. if get an error that says the 'adduser' command is not found, try '/usr/sbin/adduser'

To further protect the root account you should create an ordinary, non-root account for general use. You should only use the root account when it is necessary. You should also deny the root account the ability to login via SSH. This means that you will always need to login using your general account. When you need to do something that requires root permissions, you can become root by issuing the su command, and entering the root password.

If an attacker manages to compromise your usual user account, they can access the system but they still won't have access to the root account. An attacker would need to compromise both passwords in order to take over the system.

## Disallow root login

Once you have created a general user account, you can stop the root account from logging on directly. This is done by editing the 'sshd\_config' file, normally in /etc/ssh. Open the file in a text editor such as vi or nano (you will need to lookup instructions for using each of these programs), and look for the following line:

**PermitRootLogin yes**

Change this to:

**PermitRootLogin no**

And save the file. You will now need to login via your general account first, and then use the su command to become root. You should test that you can access your server by using your general account before you do this to make sure you do not lock yourself out of the server.

## Change your password

1. get an ssh shell on your server
2. type 'passwd' and follow the prompts.

## Change your server's hostname

1. get an ssh connection to your server
2. type 'hostname <new host name>'

This sets the hostname on your server. For other people to be able to access your server at the new name, you need to set up a DNS entry (see below).

## Set a DNS entry for your server

DNS stands for Domain Name System. It allows people to access web sites and other servers by name rather than having to remember the IP address. In order to set a DNS entry for your server you will need first to own a domain. If you have a domain with Tagadab you can set a DNS entry for your server at the domain management system here in the Tagadab Control Panel (you will need to know your server's IP address, see above). If your domain is not with Tagadab you will need to follow the instructions of the provider who hosts your domain. You can transfer domains from other providers to Tagadab by using our domain ordering page at <http://www.tagadab.com/domains>.

## Set your server's reverse DNS entry

A reverse DNS entry allows other servers which your server connects to to lookup the name of the server. This is sometimes important, for example when running a mail server, because other mail servers may check your server's name before allowing it to pass email. You can set the reverse DNS entry for all the IP addresses allocated to your server on the Tagadab control panel in the Dedicated Servers section.

## Start a website

By default your Linux server has the Apache 2 web server installed. Depending on the version of Linux you selected, this may or may not already be running. You can access the website with your browser by typing your server's IP address (or name if you have set a DNS entry). Of course there won't be much there until you have uploaded some content - see SFTP / FTP access below. If you are unable to view anything when you do this, you will need to start Apache. Login to your server via SSH, and use su to become root. Then run the following command:

```
/etc/init.d/httpd start
```

Or on Ubuntu:

```
/etc/init.d/apache2 start
```

## Enable FTP access

You can use SFTP / SCP to upload files instead of using FTP. SFTP / SCP is much more secure, as all of the content (including your password) is encrypted before being sent to the server. FTP sends all passwords and data in clear text, which is susceptible to eavesdropping.

SFTP / SCP works almost exactly the same way as FTP, although some web design packages won't support it and you will need a separate client. SFTP / SCP can be used directly from a Linux or Mac shell, or by downloading a client for Windows. We recommend WinSCP [<http://winscp.net>]. SFTP / SCP also has the advantage that you don't need to configure a separate program – if you can login via SSH you can use SFTP / SCP.

By default, all of your web pages should be uploaded to the `/var/www/` folder. This is called the web servers document root. This can be moved to another folder if desired by editing the apache configuration file. If you would still prefer to use FTP rather than SFTP / SCP, an FTP server is installed by default. However it is not enabled by default and will need to be configured. Vsftpd is installed on all of our Linux distributions.

Instructions for setting up Vsftpd are available here:

<http://mirror.centos.org/centos/3/docs/html/rhel-rg-en-3/s1-ftp-vsftpd-conf.html>

And here: <http://vsftpd.beasts.org/>.

## Additional Apache configuration

You can make changes to the way your Apache web server behaves by making changes to the Apache configuration file. This is found at:

**`/etc/httpd/conf/httpd.conf`**

Or for Ubuntu:

**`/etc/apache2/apache2.conf`**.

Full Documentation for Apache can be found at [www.apache.org](http://www.apache.org), or HowTo instructions can be found on HowToForge.

## Installing additional software

There is enough software preinstalled with your server to run most types of website. More advanced sites may require additional software, which can be installed with the build in package manager. On Fedora, Redhat and Centos this is called Yum, whilst on Ubuntu it's called apt-get. For instructions on using these, type 'man yum' or 'man apt-get' when logged in to your server.

## More information

You can get much more detailed instructions from the website of the linux distribution you have selected:

Ubuntu: <http://www.ubuntu.com>

Fedora Core: <http://fedoraproject.org/>

Red Hat: <http://www.redhat.com>

CentOS: <http://www.centos.org>

Some more general documentation can be found at The Linux Documentation project (<http://tldp.org/>) and HowToForge (<http://www.howtoforge.com/>) which is particularly good if you are new to Linux). You can also get documentation when you are connected to your server by typing in 'man' followed by the name of the program you want documentation for.