



## **Setting up an AWS Ubuntu Instance and Installing Webmin and FileMaker Server**



# Setting up an AWS Ubuntu Instance and Installing Webmin and FileMaker Server

by Oliver Reid

Webmin is a free browser-based tool for managing Linux servers

It provides a powerful and straightforward graphical interface that allows us to navigate the file system, upload and download files, manage ownership and permissions, set up cron jobs, manage users and groups, and much more.

It is an ideal tool for a user not deeply schooled in using the Linux command line, although it does have a built-in terminal module for running Linux OS commands.

This document explains "step-by-step" how to set up an Amazon Web Services Ubuntu Virtual machine (an "instance") and install Webmin and FileMaker Server.

The level of detail in this document may seem daunting, but in fact, the whole process takes about 20 minutes (not counting the time taken for the FileMaker Server download).

1. Linux User Accounts, Permissions, and Directories Overview	2
2. Linux Repositories and Package Managers	5
3. Setting up an AWS Ubuntu Server - step by step	6
4. Connecting to the AWS Ubuntu Server	11
5. Installing Webmin	14
6. Installing FileMaker Server	17
7. Saving the AWS instance as an "AMI"	23
8. Securing Webmin with an SSL certificate	24
9. Uploading and Downloading FileMaker Server files using Webmin	28

## 1. Linux User Accounts, Permissions and Directories Overview

### Users Created on Initial Setup

When you create a new Ubuntu virtual computer (instance) on Amazon Web Services (AWS), two user accounts are created by default: the *root* account and the *ubuntu* account. However, these accounts do not have passwords set.

To connect to your instance securely, you'll need to generate an RSA key pair and store the public key on your local computer. You can then use a terminal application and SSH ("Secure Shell") to connect to your instance using the *ubuntu* account.

The root account is the most powerful account on your Ubuntu instance and is a member of the "root" group. This gives the root account control over all files and actions on the computer. For security reasons, it's generally recommended to only have one account in the root group.

The *ubuntu* account is a member of the "sudo" group, which means it has almost the same capabilities as the root account, as long as you precede any commands with "sudo". The *ubuntu* account is typically used for day-to-day tasks.

You can create additional user accounts on your Ubuntu instance using the *ubuntu* account and its sudo power. You can group these accounts together and manage them as needed.

Many tools that you use to manage your Ubuntu instance, such as 'apt' for installing new programs, require sudo privileges to work.

### About Ownership and Permissions

Linux ownership and permissions are essential concepts in managing access to files and directories on a Linux system. These permissions control who can read, write, and execute files, ensuring security and proper functioning of the system.

#### Ownership:

In Linux, every file and directory has an owner and a group associated with it. The owner is the user who has created the file or directory or has been assigned ownership. The group represents a set of users who share the same access permissions. Usually, the owner is a member of this group, but this is not required.

## Permissions:

Permissions are set separately for

User (u): The file owner.

Group (g): The group the file belongs to.

Others (o): All other users on the system.

There are three types of permissions for each file or directory:

Read (r): Permission to read the contents of a file or list the contents of a directory.

Write (w): Permission to modify the contents of a file or create/delete files within a directory.

Execute (x): Permission to run a file as a program or enter and search through a directory.

These permissions are represented numerically as:

Read (r): 4

Write (w): 2

Execute (x): 1

The permissions for user, group, and others are combined to form a 3-digit number.

For example, 755. Here, the owner has read, write, and execute permissions ( $7 = 4+2+1$ ), the group has read and execute permissions ( $5 = 4+0+1$ ), and others have read and execute permissions ( $5 = 4+0+1$ ).

When executing transactions or running scripts on a Linux system, it is crucial to be aware of the ownership and permissions of the files involved.

### **Important Note About FileMaker Server**

FileMaker Server runs as the user *fmserver* which is a member of the group *fmsadmin*. Files uploaded via the 'Sharing' panel in FileMaker Pro will be assigned ownership and permissions accordingly.

However, when migrating an application from FileMaker Server, externally stored container files will need to be loaded directly to the data folder. This is easy to do using Webmin, but you will need to change the permissions on any files or folders that are not *.fmp12* files to make sure FileMaker Server can read them and write to them. This applies to the RC\_Data\_FMS, Documents, and Scripts folders, and their contents, for example. (Please see section 9: "Uploading and Downloading FileMaker Server files using Webmin".)

See: <https://help.claris.com/en/server-help/content/hostdb-upload-manual.html>

Fortunately, changing ownership and permissions is quite easy if Webmin is installed.

## Root Level Directories

At the root level in Ubuntu, we will see a number of directories. We will see these names regularly, so here is a partial list, with their functions.

<b>bin</b>	contains essential binary executable files that are required for the system to function properly.
<b>boot</b>	contains the files required for the boot process, including the kernel, initial ramdisk, and bootloader configuration files.
<b>dev</b>	contains device files, which are used by the system to interact with hardware devices
<b>etc</b>	contains system-wide configuration files.
<b>home</b>	contains user home directories
<b>opt</b>	used for installing optional software packages: this is where FileMaker Server will reside.
<b>proc</b>	provides information about running processes and system resources
<b>root</b>	the home directory of the root user
<b>run</b>	contains system runtime data, such as pid files and Unix domain sockets
<b>sbin</b>	contains system binary executables that are generally used by the system administrator.
<b>snap</b>	contains snap packages, which are self-contained software packages.
<b>srv</b>	used for serving data for services, such as websites and FTP servers
<b>sys</b>	contains system-specific files and provides an interface to the kernel
<b>tmp</b>	used for storing temporary files
<b>usr</b>	contains user programs and data files
<b>var</b>	contains variable data files, such as log files, spool files, and temporary files. In addition <code>`var/www`</code> is the default root directory for web servers on Ubuntu, and typically contains the content folders for websites

The files for FileMaker Server are stored in the ``opt`` directory.

## 2. Linux Repositories and Package Managers

A **repository**, in the context of a Linux server, is a remote location where software packages are stored and managed for installation and updates.

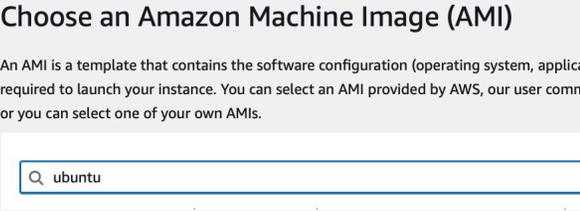
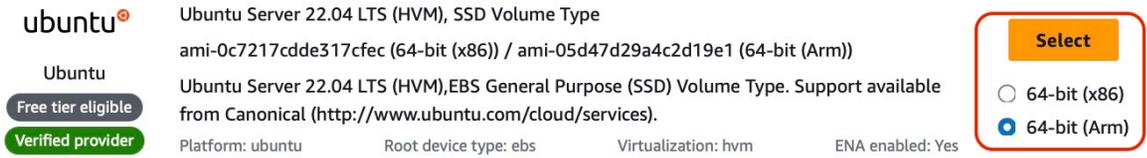
The **package manager** is a software tool that automates the process of installing, updating, configuring, and removing software packages on a computer or server. Ubuntu uses a package manager named "**apt**" (Advanced Packaging Tool).

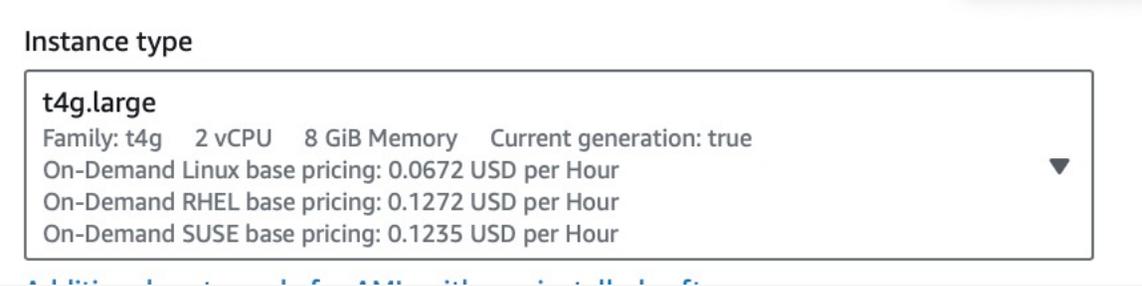
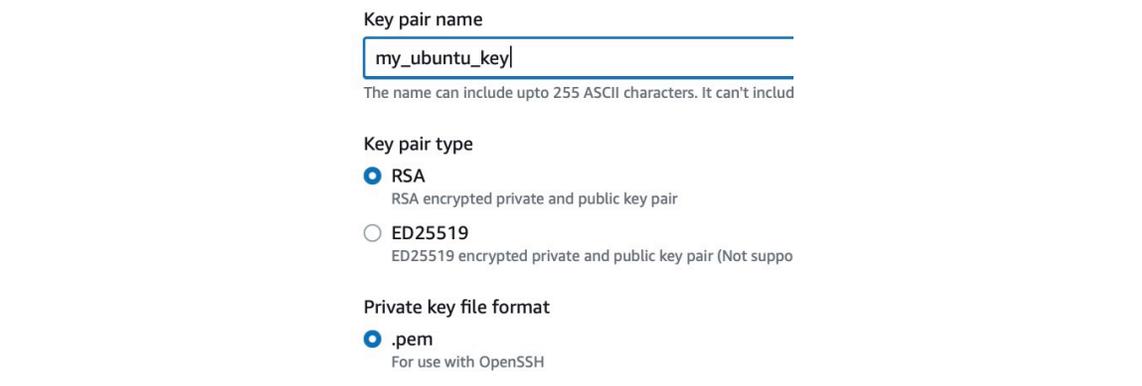
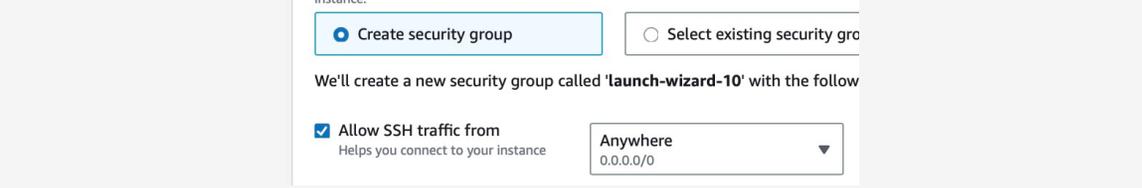
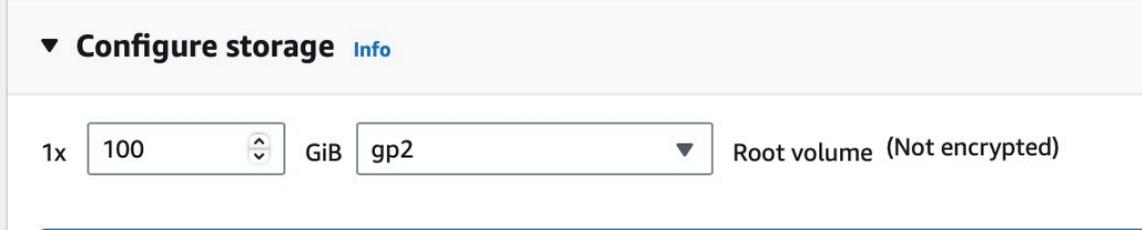
When you set up access to a new repository, the package manager stores information about the repository in a configuration file. This configuration file contains information such as the repository's name, URL, and GPG key, as well as metadata about the packages in the repository.

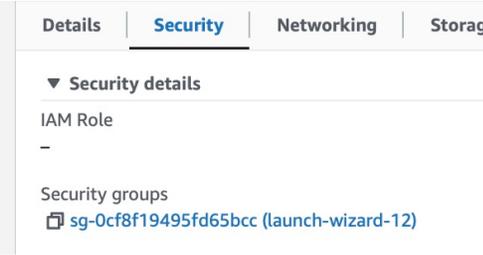
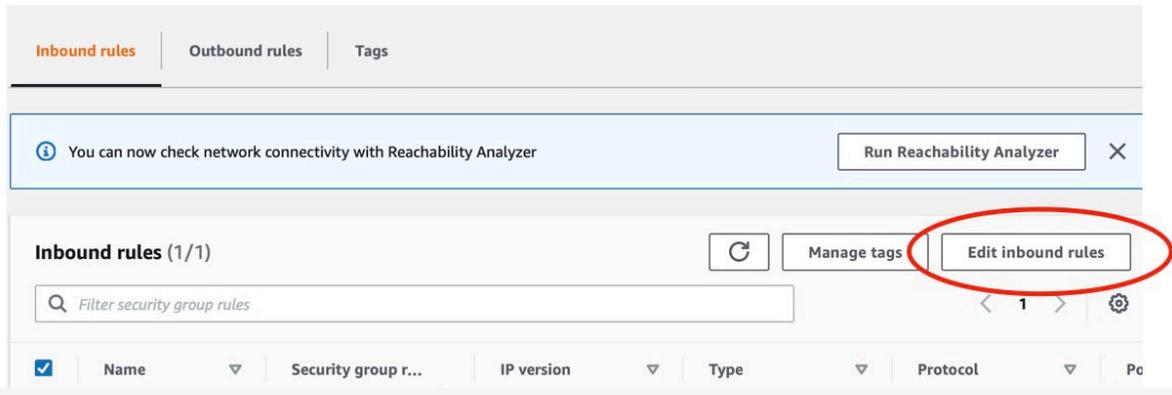
This metadata includes the package name, version number, description, dependencies, and other details needed to manage the package.

The name of the repository is used as a unique identifier, while the URL specifies the remote location of the repository's package files. The GPG key is used to verify the authenticity of the packages downloaded from the repository.

### 3. Setting up an AWS Ubuntu Server - step by step

<p>1 Go to the AWS EC2 Instances page and click on</p>	
<p>On the next screen the first task is to chose a name for your server</p>	
<p>2 Click on (lower right)</p>	
<p>3 Filter the list for `ubuntu`</p>	
<p>4 The latest version of FM Server runs on ARM processors, so that is what we will choose: Select:  Ubuntu Server 22.xx * "64-bit(Arm)"</p>	 <p>*Check the latest FileMaker specification for which Ubuntu version to use: <a href="https://support.claris.com/s/article/FileMaker-Server-operating-system-requirements-all-versions-1503692927810?language=en_US">https://support.claris.com/s/article/FileMaker-Server-operating-system-requirements-all-versions-1503692927810?language=en_US</a></p>

6	<p>Then select an instance type: you need 2 CPUs and 8GB of memory, e.g, t4g.large:</p>	
7	<p>Create a key pair (or select one you already have) A new key pair file e.g., `my_ubuntu_key.pem` will be placed in your Downloads folder  Store this carefully as you cannot download it again.</p>	
8	<p>Create a Security Group, or select one you already have. (Later we will edit this group.)</p>	
9	<p>Configure Storage. At least 100GB is recommended for FileMaker Server</p>	

10	Click on	
<p>11 Navigate back to your EC2 instance list. Your instance will take a few minutes to start up, and then you will see:</p> <p><input type="checkbox"/> My New Ubuntu Server    <a href="#">i-08eef9817e8757bbd</a>    <span style="color: green;">✔ Running</span>        t3.large    <span style="color: green;">✔ 2/2 checks passed</span></p>		
12	<p>Click on the Instance ID (in blue ). An "Instance Summary" Screen will appear. In the Security tab click on the security group name (in blue)</p>	
13	Click on "Edit Inbound Rules"	

14

We will need the ports  
 FileMaker Server requires  
 (80, 443, 5003)

Plus 22, 1022 for 'SSH'  
 access

And 10000 , which is  
 needed for Webmin to  
 function.

'All ICMP - IPv4' allows us to  
 'ping' the server

Type	Info	Protocol	Info	Port range	Info	Source	Info	
Custom TCP	▼	TCP		10000		Custom	▼	
							0.0.0.0/0	✕
HTTPS	▼	TCP		443		Custom	▼	
							0.0.0.0/0	✕
Custom TCP	▼	TCP		1022		Custom	▼	
							0.0.0.0/0	✕
HTTP	▼	TCP		80		Custom	▼	
							0.0.0.0/0	✕
SSH	▼	TCP		22		Custom	▼	
							0.0.0.0/0	✕
Custom TCP	▼	TCP		5003		Custom	▼	
							0.0.0.0/0	✕
All ICMP - IPv4	▼	ICMP		All		Custom	▼	
							0.0.0.0/0	✕

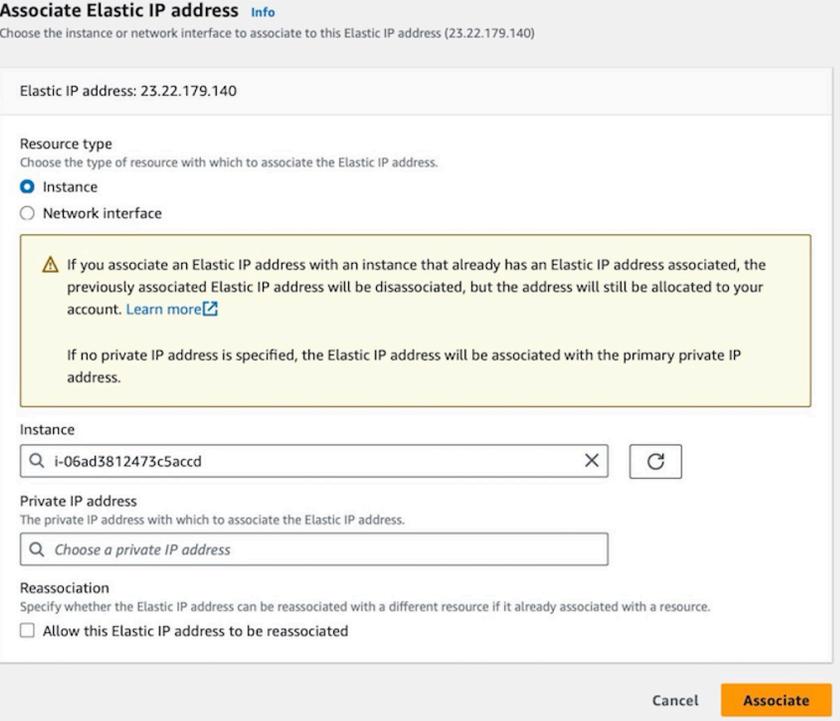
**IMPORTANT!**

**For a production system do not leave port 10000 widely accessible (0.0.0.0/0)  
 Restrict it to specific IP addresses.**

15

In the EC2 Sidebar Click on



16	Then on (upper right):	
17	<p>An Elastic IP address is an ip address which can be attached to an instance and it is (despite the name) fixed. The instance will be accessible via its Elastic IP address, and its address will not change if the instance is stopped and restarted.</p>	
18	A new Elastic IP will be created. Select it and Click on	
19	<p>Select the new instance and then click on 'Associate'</p> <p>The IP address will be attached to the instance.</p>	

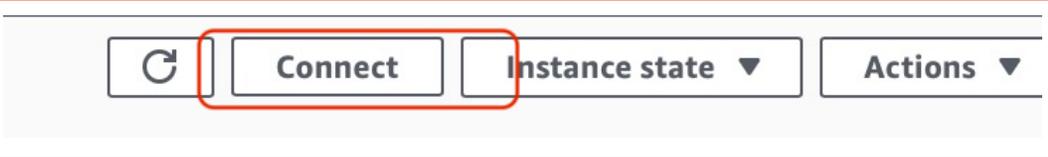
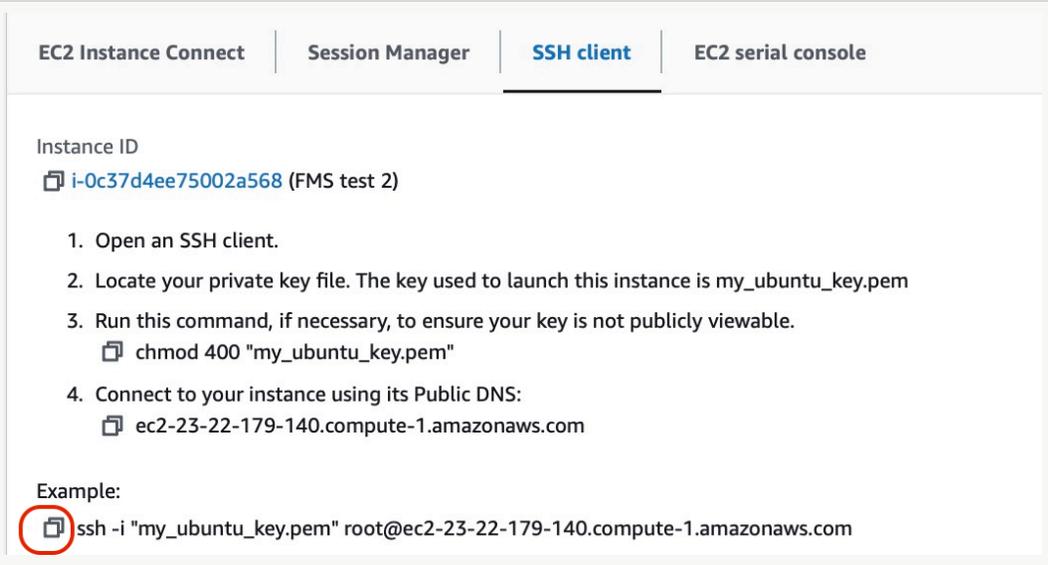
Now our server is running, and we can connect to it...

#### 4. Connecting to the AWS Ubuntu Server

To connect, we use our local Terminal application and the key file that was created when setting up the instance.

This document is illustrated using the Apple "Terminal" app. On Windows, use the "Windows PowerShell" app.

Responses entered by the user are in red type, and are on a separate line, to make copying the text easier. Note if the red type text wraps into two lines in this pdf, paste it into a plain text editor and remove any line breaks before inserting it in your terminal app.

1	Select the Ubuntu AWS instance in The EC2 instance list and click "Connect"	
2	Copy the suggested connection string	 <p>EC2 Instance Connect   Session Manager   <b>SSH client</b>   EC2 serial console</p> <p>Instance ID i-0c37d4ee75002a568 (FMS test 2)</p> <ol style="list-style-type: none"><li>1. Open an SSH client.</li><li>2. Locate your private key file. The key used to launch this instance is my_ubuntu_key.pem</li><li>3. Run this command, if necessary, to ensure your key is not publicly viewable. chmod 400 "my_ubuntu_key.pem"</li><li>4. Connect to your instance using its Public DNS: ec2-23-22-179-140.compute-1.amazonaws.com</li></ol> <p>Example: ssh -i "my_ubuntu_key.pem" root@ec2-23-22-179-140.compute-1.amazonaws.com</p>

<p>3 When we open a Terminal window on a local computer, the "current directory" is usually set to be the local user's home folder ('oliver2' in this case)</p>	<pre>Last login: Mon Apr  3 12:21:21 on ttys000 oliver2@Olivers-MacBook ~ %</pre>
<p>4 The Terminal needs to access the folder where the key file is stored. If we have stored it in (say) a folder 'key_files' we use the 'cd' command to point us to that folder</p>	<pre>Last login: Mon Apr  3 12:21:21 on ttys000 oliver2@Olivers-MacBook ~ % cd key_files oliver2@Olivers-MacBook key_files %</pre>
<p>5 Pasting in the `ssh` command string, copied above, command creates a ssh (secure shell) connection to the server.</p> <p><i>Note:</i> if AWS suggests "root@...." as the username, change it to "ubuntu@...."</p>	<pre>ssh -i "my_ubuntu_key.pem" ubuntu@ec2-23-22-179-140.compute-1.amazonaws.com</pre>
<p>Note: You may receive a message saying the <i>my_ubuntu_key.pem</i> has access permissions that are too open, and the connection fails.</p> <p>In that case run: <code>sudo chmod 400 "my_ubuntu_key.pem"</code></p>	

<p>Enter 'yes' at this prompt:</p>	<pre>oliver2@Olivers-MacBook key_files % ssh -i "my_ubuntu_key.pem" ubuntu@ec2-23-22-179-140.compute-1.amazonaws.com The authenticity of host 'ec2-23-22-179-140.compute-1.amazonaws.com (54.146.105.206)' can't be established. ED25519 key fingerprint is SHA256:I3ekj8Bn8JrcHPWtkKMD6UkUNCpxVZ204Imp4LEzDg. This key is not known by any other names Are you sure you want to continue connecting (yes/no/ [fingerprint])? yes</pre>
<p>6 ..and the connection will proceed Eventually we see a prompt from the Ubuntu server. We can now execute Linux commands on the server</p>	<pre>Are you sure you want to continue connecting (yes/no/ [fingerprint])? yes Warning: Permanently added 'ec2-23-22-179-140.compute-1.amazonaws.com' (ED25519) to the list of known hosts. Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1033-aws x86_64) Last login: Mon Apr  3 19:22:00 2023 from 67.83.209.112 ubuntu@ip-172-31-0-235:~\$</pre>

Notes:

on Mac OS, "::~~\$" indicates that the system is waiting for a response.

The IP address quoted in the command prompt (in green) is AWS' *internal* IP address, not its *public* IP address, which is the Elastic IP address we associated with the Ubuntu instance.

## 4. Installing Webmin

Webmin is designed to be installed on a 'clean' Linux Instance - with no other applications installed (yet).

The Webmin site provides a 'shell' (i.e., OS level) script to set up the repository for Webmin.

(Instructions can be found here <https://webmin.com/download/> )

1	Open an SSH Connection																																										
3	Retrieve script to set up the repository - this command will do that:	<pre>ubuntu@ip-172-31-0-235:~\$ sudo curl -o setup-repos.sh https://raw.githubusercontent.com/webmin/webmin/master/setup-repos.sh Note: remove any line break if copying from here.</pre> <table border="1"> <thead> <tr> <th>% Total</th> <th>% Received</th> <th>% Xferd</th> <th>Average Speed</th> <th>Time</th> <th>Time</th> </tr> <tr> <th>Time</th> <th>Current</th> <th></th> <th>Dload</th> <th>Upload</th> <th>Total</th> <th>Spent</th> </tr> </thead> <tbody> <tr> <td>Left</td> <td>Speed</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>100</td> <td>4772</td> <td>100</td> <td>4772</td> <td>0</td> <td>0</td> <td>33370</td> </tr> <tr> <td>---</td> <td>---</td> <td>---</td> <td>---</td> <td>---</td> <td>---</td> <td>---</td> </tr> <tr> <td></td> <td>33605</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <pre>ubuntu@ip-172-31-0-235:~\$</pre>	% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current		Dload	Upload	Total	Spent	Left	Speed						100	4772	100	4772	0	0	33370	---	---	---	---	---	---	---		33605					
	% Total	% Received	% Xferd	Average Speed	Time	Time																																					
Time	Current		Dload	Upload	Total	Spent																																					
Left	Speed																																										
100	4772	100	4772	0	0	33370																																					
---	---	---	---	---	---	---																																					
	33605																																										
Then we run that script with this command	<pre>ubuntu@ip-172-31-0-235:~\$ sudo sh setup-repos.sh</pre>																																										
4	Enter 'y' at this prompt	<pre>Setup Webmin official repository? (y/N) y</pre>																																									
5	The repository setup proceeds:	<pre>Downloading Webmin key .. . Webmin package can now be installed using apt-get install webmin command.. ubuntu@ip-172-31-0-235:~\$</pre>																																									

6	We need to make sure that the `wget` and `unzip` commands are available, and that `apt` is up to date.	<pre>ubuntu@ip-172-31-0-235:~\$ sudo apt install unzip .. <u>and then</u> ubuntu@ip-172-31-0-235:~\$ sudo apt install wget .. <u>and then</u> ubuntu@ip-172-31-0-235:~\$ sudo apt update</pre>
	This command will start the installation	<pre>sudo apt-get install webmin</pre>
7	After the Webmin software components have been downloaded, reply "Y" to this prompts	<pre>.ubuntu@ip-172-31-0-235:~\$ sudo apt-get install webmin . Do you want to continue? [Y/n] Y . .</pre>
	The installation may take a minuter or so.. Eventually you will see:	<pre>. . . Processing triggers for man-db (2.9.1-1) ... Processing triggers for mime-support (3.64ubuntu1) ... ubuntu@ip-172-31-0-235:~\$</pre>

<p>8 We need to create user name to access Webmin.</p> <p>Decide on suitable user name (we are using 'my_user' in this example) and create it using the `adduser` command.</p> <p>`adduser` will create both a user and group with the name you have chosen and prompt you to enter a password.</p>	<pre>ubuntu@ip-172-31-0-235:~\$ sudo adduser my_user Adding user `my_user' ... Adding new group `my_user' (1001) ... Adding new user `my_user' (1001) with group `my_user' ... Creating home directory `/home/my_user' ... Copying files from `/etc/skel' ... New password: &lt;enter your password here&gt; Retype new password: &lt;enter your password here&gt; passwd: password updated successfully Changing the user information for my_user Enter the new value, or press ENTER for the default   Full Name []: Oliver Reid   Room Number []:   Work Phone []:   Home Phone []:   Other []: Is the information correct? [Y/n] Y ubuntu@ip-172-31-0-235:~\$</pre>
<p>9 Then we add sudo group membership to the new user using the `usermod` command with the a and G options</p>	<pre>ubuntu@ip-172-31-0-235:~\$ sudo usermod -aG sudo my_user</pre>
<p>10 We can now use this account to login and view the Webmin interface in a browser using the url http://&lt;server external ipaddress&gt;:10000</p> <p><b>Note:</b> Use <b>http, not https</b>, until an SSL certificate for Webmin has been set up. Browsers may implicitly add https:// if you use the IP address alone, which will cause an error.</p> <p>IMPORTANT: the IP address shown above is the AWS <i>Internal</i> address. You should in fact use public IP address to reach Webmin via a browser - the Elastic IP address that is attached to the Ubuntu instance</p>	

Click to go back, hold to see history

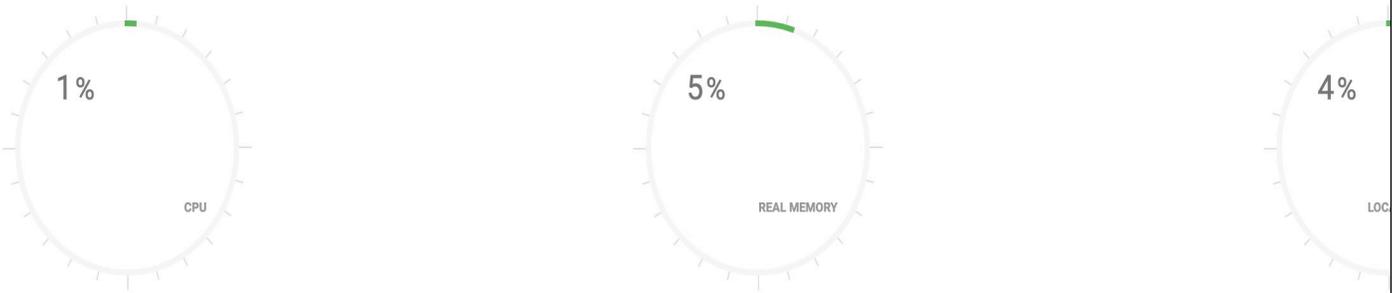
Webmin Dashboard

Search

- Webmin
- System
- Servers
- Tools
  - Command Shell
  - Custom Commands
  - File Manager
  - HTTP Tunnel
  - Perl Modules
  - Protected Web Directories
  - System and Server Status
  - Terminal
  - Upload and Download
- Networking
- Hardware
- Cluster
- Un-used Modules
- Refresh Modules

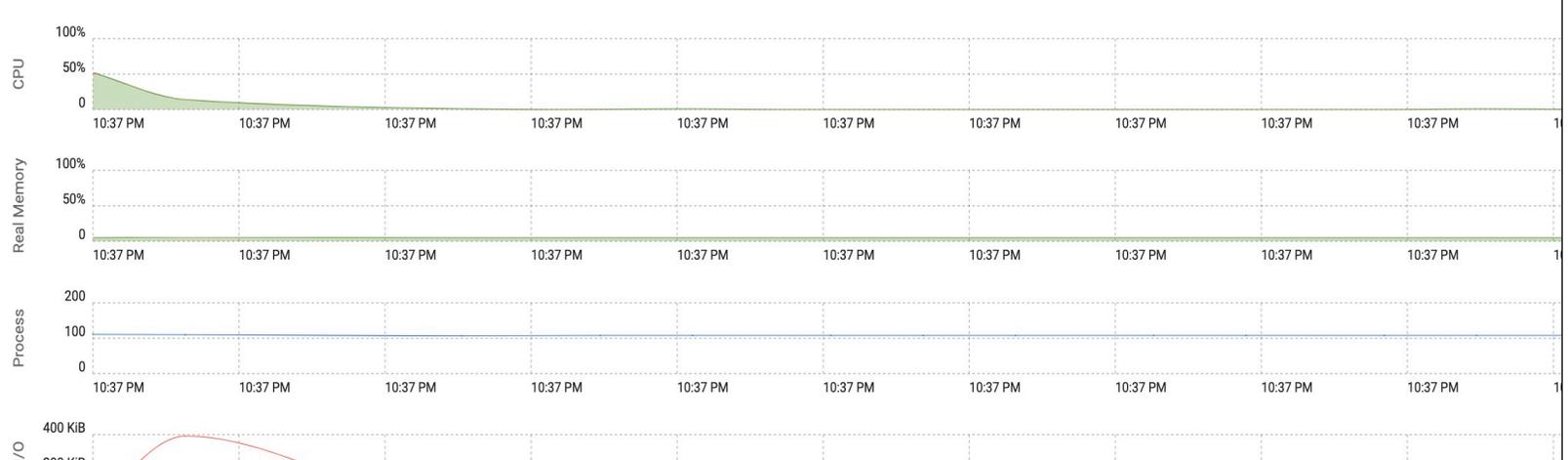
my\_user

### System Information



System hostname	<a href="#">ip-172-31-89-12.ec2.internal (172.31.89.12)</a>	Operating system	Ubuntu Linux 20.04.6
Webmin version	2.021	Authentic theme version	20.21
Time on system	Monday, April 3, 2023 10:37 PM	Kernel and CPU	Linux 5.15.0-1033-aws on x86_64
Processor information	Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz, 2 cores	System uptime	54 minutes
Running processes	107	CPU load averages	0.10 (1 min) 0.05 (5 mins) 0.05 (15 mins)
Real memory	330.57 MiB used / 931.86 MiB cached / 7.66 GiB total	Local disk space	2.03 GiB used / 46.34 GiB free / 48.37 GiB total
Package updates	3 package updates are available		

### Stats History



## 5 Installing FileMaker Server

The FileMaker Server installation process makes changes to the server configuration:

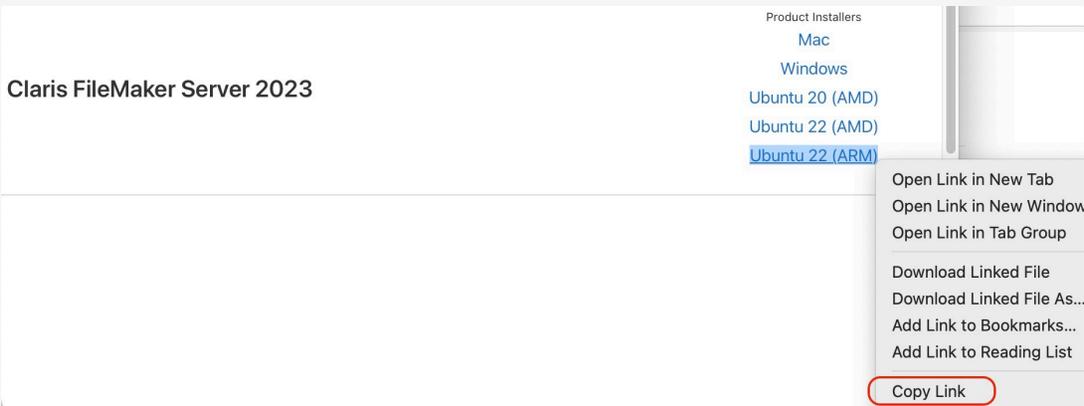
- Apache is installed but then it is disabled.
- The Nginx web server is installed and the web root folder is set to a folder inside the FileMaker Directory:  
/opt/FileMaker/FileMaker Server/NginxServer/htdocs
- You can switch to using Apache - see:  
[https://support.claris.com/s/answerview?anum=000035914&language=en\\_US#a5](https://support.claris.com/s/answerview?anum=000035914&language=en_US#a5)
- The standard Ubuntu firewall software, UFW, is disabled and FirewallD is set up instead. *Only the ports used by FileMaker Server and SSH are opened (port 10000 is not opened).*

Webmin provides a Terminal window for using Linux commands. However, this should not be used to install FileMaker Server, because Webmin will disconnect as soon as port 10000 becomes unavailable.

So we use the local Terminal app to install FileMaker Server. *We can reopen port 10000 again once the installation is complete.*

The steps to install FileMaker Server are:

1	These two commands will create directory with the 'ubuntu' user home directory, to store the installation files, and make this the current directory.	<pre>ubuntu@ip-172-31-0-235:~\$ mkdir fminstaller ubuntu@ip-172-31-0-235:~\$ cd fminstaller ubuntu@ip-172-31-0-235:~/fminstaller</pre>
---	---	--

2	<p>Copy the url on you downloads page for the FileMaker Server installer for the version on Ubuntu you are using</p>	 <p>Claris FileMaker Server 2023</p> <p>Product Installers  Mac  Windows  Ubuntu 20 (AMD)  Ubuntu 22 (AMD)  Ubuntu 22 (ARM)</p> <p>Open Link in New Tab  Open Link in New Window  Open Link in Tab Group  Download Linked File  Download Linked File As...  Add Link to Bookmarks...  Add Link to Reading List  Copy Link</p> <p>It will be something like: <a href="https://downloads.claris.com/esd/fms_20.3.1.31_Ubuntu22_arm64.zip">https://downloads.claris.com/esd/fms_20.3.1.31_Ubuntu22_arm64.zip</a></p>
3	<p>We will use the <code>wget</code> command (<i>note the <code>`sudo`</code> before <code>`wget...`</code></i>) to download the installation package: it will be saved in the <code>fminstaller</code> directory we created.</p> <p>It may take a while — but usually less than 15 mins</p>	<pre>ubuntu@ip-172-31-0-235:~/fminstaller\$ sudo wget https://downloads.claris.com/esd/ fms_20.3.1.31_Ubuntu22_arm64.zip</pre>

4	<p>When the download is complete, unzip the package.</p> <p>Note: the unzipped file is <i>filemaker-server-20.3.1.31-arm64.deb</i> (not <i>fms-40.1.2.8...</i>)</p> <p><b>NOTE:</b> Update the version number to match the version number in the download link</p>	<pre>ubuntu@ip-172-31-0-235:~/fminstaller\$ sudo unzip fms_20.3.1.31_Ubuntu22_arm64.zip  archive: fms_20.3.1.31_Ubuntu22_arm64.zip   inflating: filemaker-server-20.3.1.31-arm64.deb   inflating: Assisted Install.txt   inflating: README_Installation_English.txt   inflating: README_Installation_French.txt   inflating: README_Installation_German.txt ..... ubuntu@ip-172-31-0-235:~/fminstaller\$</pre>
	<p>Run the 'ls' command to list the directory contents so we can see the name of the unzipped installer file. (it has the extension <i>.deb</i>)</p>	<pre>ubuntu@ip-172-31-0-235:~/fminstaller\$ ls 'Assisted Install.txt' 'FMS License (Japanese).rtf' README_Installation_Italian.txt . . . . filemaker-server-20.3.1.31-arm64.deb</pre>
5	<p>The <i>apt install</i> command can now install FileMaker Server:</p>	<pre>ubuntu@ip-172-31-0-235:~\$ sudo apt install ./filemaker-server-20.3.1.31-arm64.deb</pre> <p>NOTE `apt install`.. does not automatically look for a file in the current directory: the `./` preceding the file name directs the operation there.</p>

6	Be ready to respond to prompts during the install, e.g.:	<pre> Reading package lists... Done Building dependency tree... Done Reading state information... Done  . . . . Need to get 607 MB/1010 MB of archives. After this operation, 3301 MB of additional disk space will be used. Do you want to continue? [Y/n] Y ....     I confirm that I have read and agree to the     terms of the Claris FileMaker Server Software License Agreement     included with the software.     Agree(y)          Decline(n) [y/n] y     0) Claris FileMaker Server primary machine     1) Claris FileMaker WebDirect secondary machine     Choose 0 to install Claris FileMaker Server primary machine,     or 1 to install Claris FileMaker WebDirect secondary machine.     [0/1] 0     ...     Set up the Claris FileMaker Server Admin Console account for     Claris FileMaker Server primary machine.     Use this account when you sign into Claris FileMaker Server     Admin Console.     Enter User Name: &lt;choose and enter a user name&gt;     Create a password to sign into Claris FileMaker Server Admin     Console.     Enter Password:&lt;enter password&gt;     Confirm Password: &lt;confirm password&gt;     Create a 4-digit PIN needed to reset Claris FileMaker Server     Admin Console account password via the command line interface.     Enter PIN: &lt;enter PIN&gt;     Confirm PIN: &lt;enter PIN&gt;     ... </pre>
---	--	--

7	When the install is finished you will see	<p>To configure FileMaker Server, open Admin Console at:  <a href="https://172.31.49.205/admin-console">https://172.31.49.205/admin-console</a>  .....  ubuntu@ip-172-31-0-235:~\$</p>
<p>IMPORTANT: the IP address cited here is the AWS <i>Internal</i> address. You should in fact use the public IP address - the Elastic IP address that was attached to the ubuntu instance</p>		

**Now, Webmin will now not be accessible because port 10000 is no longer open!**

The FM Server install disables the standard firewall service (*ufw*) and sets up *firewalld* instead

Actually, we do not need the Ubuntu firewall to be running, as the AWS Security Group that we set up serves the same security need.

8	<b>So we disable the firewall:</b>	<pre>ubuntu@ip-172-31-0-235:~/fminstaller\$ sudo systemctl stop firewalld ubuntu@ip-172-31-0-235:~/fminstaller\$ sudo systemctl disable firewalld Removed /etc/systemd/system/multi-user.target.wants/firewalld.service. Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service..... ubuntu@ip-172-31-0-235:~/fminstaller\$</pre> <p>Note: the second command <code>sudo systemctl disable firewalld</code> ensures that <i>firewalld</i> will not start up again if the instance is rebooted.</p>
---	------------------------------------	---

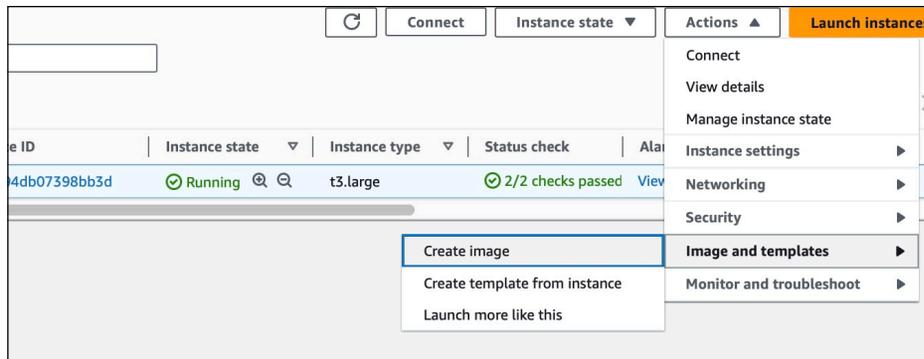
## 7 Saving the instance as an "AMI"

Save an AWS Image (AMI) of this instance at this point. You can launch additional instances from this image for future installations.

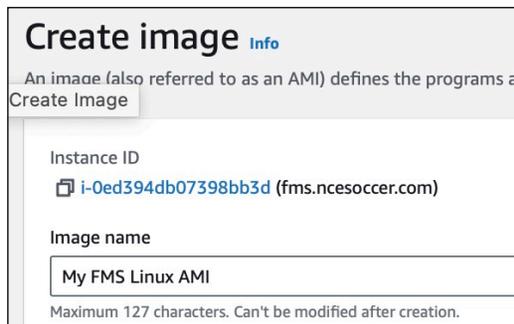
*DO THIS BEFORE SETTING UP SSL* (below). Otherwise you may have trouble connecting to an AMI-based instance that but has a different IP address

To create an AMI using the AWS console:

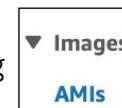
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Select the instance and then select 'Create Image' from the Actions Menu



3. Provide a name for your AMI: and click 'Create Image'

A screenshot of the 'Create image' form in the AWS console. The form title is 'Create image' with an 'Info' icon. Below the title is a description: 'An image (also referred to as an AMI) defines the programs and operating system on your Amazon EC2 instances.' The form contains two input fields: 'Instance ID' with a dropdown menu showing 'i-0ed394db07398bb3d (fms.ncesoccer.com)' and 'Image name' with a text input field containing 'My FMS Linux AMI'. A note below the image name field states: 'Maximum 127 characters. Can't be modified after creation.'

You can launch an instance from your AMI by selecting it after clicking



## 8 Securing Webmin with an SSL certificate

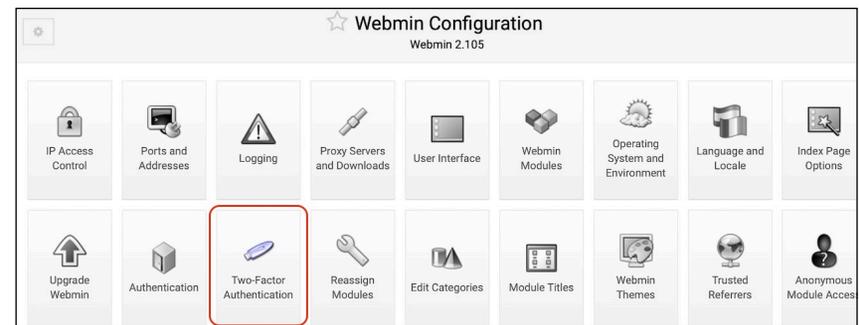
Webmin runs using a dedicated web server called "Miniserv". This means you can use Webmin to manage servers such as Nginx and Apache from Webmin, without interfering with the operation of Webmin itself.

You can install a Webmin "self-signed" certificate for Miniserv to use, in a few clicks. Doing this ensures that communication with the server via Webmin, including login credentials, and files transferred, are encrypted.

**NOTE:** Some browsers may still warn that there is a security risk .. but traffic is still encrypted if a self-signed certificate is in place.

You can also edit the AWS security group to restrict access via port 10000, which Webmin uses, to a single machine, or group of machines in your office. This level of security may well suffice. You can also set up 2-factor authentication here:

To set up a self-signed certificate:



1	Select "Webmin Configuration "	A screenshot of the Webmin navigation menu. The menu is dark blue with white text. The items are: "Webmin" (with a gear icon), "Backup Configuration Files", "Change Language and Theme", "Webmin Actions Log", "Webmin Configuration" (highlighted with a red oval), and "Webmin Servers Index".
2	Click on "SSL Encryption"	

3	<p>Click "Self-Signed Certificate"</p>
4	<p>Complete the form</p> <p>Use the IP address (or domain if you have set up a DNS A record for this address)</p> <p>Enter "US" (or your country code)</p> <p>Check "Use new key immediately"</p>
5	<p>Click "Create Now"</p> <p></p> <p>You may get an alert saying the server could be accessed. In that case simply reconnect with the starting url: e.g.,</p> <p><code>https://23.22.179:10000</code></p> <p>Note: you can now use 'https:// .....'</p>

[SSL Settings](#)
[Current Certificate](#)
[Per-IP Certificates](#)
[Self-Signed Certificate](#)
[Certificate Signing Request](#)
[Upload Certificate](#)
[Let's Encrypt](#)

This form can be used to create a new SSL key and certificate for your Webmin server.

**Create SSL key**

**Server names**  Any hostname  54.146.105.206  
**Email address**   
**Department**   
**Organization** Webmin Webserver on ip-172-31-89-12.e  
**City or locality**   
**State**   
**Country code**   
**SSL key size**  Default (2048)   bits  
**Days before expiry** 1825  
**Write key to file** /etc/webmin/miniserv.pem  
**Use new key immediately?**  Yes  No

**Create Now**

6	Select the "Current Certificate" tab	You see that the certificate has been applied, and expires in 5 years
9	Enable SSL	

**IMPORTANT:** if you plan to relaunch this instance with a different IP address, disable SSL first.

**Enable SSL?**  
 Yes  No

Otherwise, you may have difficulty accessing the instance again:

You can also install a custom SSL certificate here (but this requires adding a 'reverse proxy' to nginx):

Certificate Signing Request
Upload Certificate

## 9. Uploading and Downloading FileMaker Server files using Webmin

Webmin provides a very useful facility "Upload and Download".

*Importantly, it allows you to choose the destination directory and set the intended ownership of the uploaded files in one step.*

You can also upload multiple files at once: a time-saver if migrating from another server.

Download from web   Upload to server   Download from server

This page allows you to upload one or more files from the PC on which your web browser runs to the system running Webmin

### Upload files to server

**Files to upload**

2 files selected

**File or directory to upload to**

/opt/FileMaker/FileMaker Server/Data/Databases  Create directory if needed?

**Owned by user**

fmserver

**Owned by group**

Default

**Extract archive or compressed files?**

Yes, then delete    Yes    No

**Send email when uploads are done?**

No    Yes, to address

Upload

