How To Install Z Shell (zsh) on a Cloud Server

*Authored by: **ASPHostServer Administrator** [asphostserver@gmail.com]*
Saved From: http://faq.asphosthelpdesk.com/article.php?id=209

Introduction

Z shell, or zsh, is a shell for Linux and other Unix-like operating systems. It is built to be a drop-in replacement for the traditional bash shell. The zsh offers modern auto-completion and auto-correction, powerful scripting capabilities, and extensibility with modules.

In this guide, we will install zsh on Ubuntu 12.04 cloud server and configure it with some useful defaults. We will then go over some of the features that might make zsh a strong contender for being your default shell.

Installation

Since zsh is in Ubuntu"s default repositories, installation is quite easy and straightforward.

```
sudo apt-get update
sudo apt-get install zsh
```

We now have zsh on our server, but nothing has happened yet. For now, we"ll have to start zsh like any other program. Later, we"ll go over how you can change your default shell to zsh once you are comfortable with it. Let"s get started.

```
zsh
```

Since this is our first time running zsh, we won"t be dropped immediately into a Z shell session. Instead, we are greeted by a menu that will help us choose some good default configuration settings. We recommend that you choose option (1).

```
(1)  Continue to the main menu.
```

You will see a new menu where you can choose some of what you"d like in your configuration file. For the first three options, you should probably follow the recommended configurations unless you have a reason to change something.

One thing that might not be clear is that you need to select each option and set it for it to be added to your configuration. The zsh-newuser-install script (the menuing system you are now viewing) will not add any configuration settings by default, even if they are recommended.

Keep in mind that it is easy to change these settings by editing the ~/.zshrc file, so don"t worry about it too much if you aren"t sure about something right now.

We"re going to choose the default options for the first three menu items, and we"re going to move on to some shell options with selection (4). So make sure you go through the first 3 menu items and set them up with the defaults.

The first three items should be marked with "(Unsaved changes.)â€• following your selections.

```
(1)  Configure settings for history, i.e. command lines remembered
     and saved by the shell.  (Unsaved changes.)
(2)  Use the new completion system.  (Unsaved changes.)
(3)  Configure how keys behave when editing command lines.  (Unsaved changes.)
```

Next, we"re going to set some options in item (4). Choose this from the menu.

```
(4)  Pick some of the more common shell options.  These are simple "on"
     or "off" switches controlling the shell's features.
```

In this sub-menu we want to turn option 1, 2, and 3 on. Do this by selecting each item and then hitting "s€• to turn it on when you are presented with the appropriate sub-menu.

The menu should look like this when you are complete.

```
# (1) Change directory given just path.
setopt autocd                                            (set but not saved)
# (2) Use additional pattern matching features.
setopt extendedglob                                      (set but not saved)
# (3) Append new history lines instead of overwriting.
setopt appendhistory                                     (set but not saved)
```

The other options in this menu are up to your personal preference. Feel free to adjust them if you would like. When you are finished, type (0) to save your changes and return to the main menu.

```
(0)  Remember edits and return to main menu (does not save file yet)
```

At this point, we are done with our initial configuration with the menu. We can type (0) to exit and save our settings.

```
(0)  Exit, saving the new settings.  They will take effect immediately.
```

We are finally dropped to the zsh for the first time. Let"s check out the configuration file generated by the zsh-newuser-install script.

```
nano ~/.zshrc
```

It looks pretty good so far. This is the file we"ll want to edit if we want to change the way that zsh works later on.

Let"s try it out now by changing our prompt. We need to either exit out of nano with ctrl-X or start a new zsh session to try out the different options we have. Once you"re back to a zsh prompt, type this.

```
autoload -U promptinit
promptinit
```

We"ve now loaded our prompt-styling mechanisms. Let"s choose one of the default styles to try out. To get a list of available styles, type this into the prompt:

```
prompt -l
```

```
adam1 adam2 bart bigfade clint elite2 elite fade fire off oliver pws redhat suse
```

```
walters zefram
```

This will give you a list of available prompt themes. Let"s try out the "clintâ€• theme.

```
prompt clint
```

You should see your prompt change to reflect your new theme. When you find one that you want, we"ll add it to our .zshrc so that it because the permanent default.

```
nano ~/.zshrc
```

At the end, add the following

```
autoload -U promptinit
promptinit
prompt clint
```

Now, the next time you start a zsh in your server, you will have your prompt configured the way you like it.

To make changes available immediately, you"ll want to source your configuration file.

```
source ~/.zshrc
```