



Patrick Dahlke

[Follow](#)

Skier, Surfer, Software Engineer, Open Source Lover, Writer, Speaker, Tech Leader

Oct 13, 2017 · 4 min read

## How to: Install Go 1.11.2 on Ubuntu 16.04

### Introduction

**Go** is an open source, modern programming language developed by Google that uses high-level syntax similar to scripting languages and makes it easy to build simple, reliable, and efficient software. It is popular for many applications, at many companies, and has a robust set of tools and over 90,000 repos.

This tutorial will walk you through downloading and installing Go 1.11.2, as well as building a simple Hello World application. It's an update/edit of my other story "How to: Install Go 1.8 on Ubuntu 16.04"

### Prerequisites

- One sudo non-root user

### Step 1—Installing Go

Let's install go1.11.2 on your PC or server

If you are ready, update and upgrade the Ubuntu packages on your machine. This ensures that you have the latest security patches and fixes, as well as updated repos for your new packages.

```
sudo apt-get update
sudo apt-get -y upgrade
```

With that complete, you can begin downloading the latest package for Go by running this command, which will pull down the Go package file, and save it to your current working directory, which you can determine by running `pwd`.

```
curl -O
https://storage.googleapis.com/golang/go1.11.2.linux-
amd64.tar.gz
```

Next, use `tar` to unpack the package. This command will use the Tar tool to open and expand the downloaded file, and creates a folder using the package name, and then moves it to `/usr/local`.

```
tar -xvf go1.11.2.linux-amd64.tar.gz

sudo mv go /usr/local
```

Some users prefer different locations for their Go installation, or may have mandated software locations. The Go package is now in `/usr/local` which also ensures Go is in your `$PATH` for Linux. It is possible to install Go to an alternate location but the `$PATH` information will change. The location you pick to house your Go folder will be referenced later in this tutorial, so remember where you placed it if the location is different than `/usr/local`.

## Step 2—Setting Go Paths

In this step, we'll set some paths that Go needs. The paths in this step are all given are relative to the location of your Go installation in `/usr/local`. If you chose a new directory, or left the file in download location, modify the commands to match your new location.

First, set Go's root value, which tells Go where to look for its files.

```
sudo nano ~/.profile
```

At the end of the file, add this line:

```
export PATH=$PATH:/usr/local/go/bin
```

If you chose an alternate installation location for Go, add these lines instead to the same file. This example shows the commands if Go is installed in your home directory:

```
export GOROOT=$HOME/go
export PATH=$PATH:$GOROOT/bin
```

With the appropriate line pasted into your profile, save and close the file. Next, refresh your profile by running:

```
source ~/.profile
```

## Step 3—Testing your go 1.11.2 installation

Now that Go is installed and the paths are set for your machine, you can test to ensure that Go is working as expected.

Easy and simplest way: type

```
go version //and it should print the installed go version
1.11.2
```

Create a new directory for your Go workspace, which is where Go will build its files.

```
mkdir $HOME/work
```

Now you can point Go to the new workspace you just created by exporting `GOPATH` .

```
export GOPATH=$HOME/work
```

For me, the perfect `GOPATH` is `$HOME`

```
export GOPATH=$HOME
```

Then, create a directory hierarchy in this folder through this command in order for you to create your test file. You can replace the value user with your GitHub username if you plan to use Git to commit and store your Go code on GitHub. If you do not plan to use GitHub to store and manage your code, your folder structure could be something different, like `~/my_project`.

```
mkdir -p work/src/github.com/user/hello
```

Next, you can create a simple “Hello World” Go file.

```
nano work/src/github.com/user/hello/hello.go
```

Inside your editor, paste in the content below, which uses the main Go packages, imports the formatted IO content component, and sets a new function to print ‘Hello World’ when run.

```
package main

import "fmt"

func main() {
    fmt.Printf("hello, world\n")
}
```

This file will show “Hello, World” if it successfully runs, which shows that Go is building files correctly. Save and close the file, then compile it invoking the Go command `install` .

```
go install github.com/user/hello
```

With the file compiled, you can run it by simply referring to the file at your Go path.

```
sudo $GOPATH/bin/hello
```

If that command returns “Hello World”, then Go is successfully installed and functional [1].

— That’s it-go1.11.2 is installed

## Conclusion

By downloading and installing the latest Go package and setting its paths, you now have a PC/machine to use for Go development.

Check out the original version from DigitalOcean, this post is a copy, but for the 1.11.2 version:

<https://www.digitalocean.com/community/tutorials/how-to-install-go-1-6-on-ubuntu-16-04>



