How To: Install Go on Raspberry Pi

There’s nothing special about installing Go language on Raspberry Pi platform, apart from the fact that you will get the version of Go compiled for the platform Raspberry Pi is using – which is ARM based.

Install Go using apt-get

Simply install the golang package, expect it to bring quite a few dependencies along:

greys@becky:~ $ sudo apt-get install golang
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libcgi-fast-perl libcgi-pm-perl libdate-manip-perl libdbi
  libdevel-globaldestruction-perl libdist-checkconflicts-perl
  libemail-date-format-perl
  libexporter-tiny-perl libfcgi-perl libfile-copy-recursive-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl
  libhttp-message-perl libio-html-perl libio-multiplex-perl
  libio-socket-inet6-perl libio-socket-ssl-perl libipc-shareable-perl
  liblist-moreutils-perl liblog-dispatch-perl liblog-log4perl
  perl liblwp-mediatypes-perl libmailtools-perl libmime-lite-
perl libmime-types-perl
libmodule-implementation-perl libmodule-runtime-perl libnet-cidr-perl libnet-libidn-perl libnet-server-perl libnet-smtp-ssl-perl libnet-snmp-perl
libnet-ssleay-perl libparams-classify-perl libparams-validate-perl librrd8 librrds-perl libsocket6-perl libsub-exporter-progressive-perl
libsub-name-perl libtry-tiny-perl lxkeymap munin-doc perl-openssl-defaults python-cairo python-gtk2 python-xklavier rrdtool
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
golang-1.7 golang-1.7-doc golang-1.7-go golang-1.7-src golang-doc golang-go golang-src
Suggested packages:
bzr mercurial
The following NEW packages will be installed:
golang golang-1.7 golang-1.7-doc golang-1.7-go golang-1.7-src golang-doc golang-go golang-src
0 upgraded, 8 newly installed, 0 to remove and 182 not upgraded.
Need to get 27.5 MB of archives.
After this operation, 144 MB of additional disk space will be used.
Do you want to continue? [Y/n] y

Run go to check version

greys@becky:~ $ go version
go version go1.7.4 linux/arm

Have fun!

See Also

- Raspberry Pi
- What Raspbian Version Do I have?
- Confirm Raspberry Pi model
Now and then you may notice that `apt-get upgrade` command keeps a few packages back, meaning they don’t get upgraded. This quick post shows what you can do about it and how to get all the packages upgraded.

**How apt-get Keeps Packages Back**

This is how keeping packages back will look like:

```bash
greys@xps:~ $ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
linux-generic linux-headers-generic linux-image-generic
0 to upgrade, 0 to newly install, 0 to remove and 3 not to upgrade.
```
Why Packages Are Kept Back by apt-get

Most likely reason for keeping packages back is that upgrading them means installing new packages or removing existing ones. apt-get upgrade strictly upgrades existing packages, without removing or installing anything.

Since the command you’re giving to apt-get is upgrade and not to install new packages, packages that require some old packages removed or new packages installed are kept back.

How To Upgrade Packages That Were Kept Back

Simply use the dist-upgrade option of apt-get, which will resolve dependencies and install/remove dependent package as needed:

```bash
greys@xps:~ $ sudo apt-get dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed
  linux-headers-5.0.0-20  linux-headers-5.0.0-20-generic  linux-image-5.0.0-20-generic  linux-modules-5.0.0-20-generic  linux-modules-extra-5.0.0-20-generic
The following packages will be upgraded:
  linux-generic linux-headers-generic linux-image-generic
3 to upgrade, 5 to newly install, 0 to remove and 0 not to upgrade.
Need to get 67.0 MB of archives.
After this operation, 334 MB of additional disk space will be used.
Do you want to continue? [Y/n] y

That’s it for todat, enjoy!
Automatically Remove Packages in Ubuntu

It is possible that removing some packages on your Debian, Ubuntu or Mint Linux system will leave a number of software packages behind. Although they are not required by any existing package, they are still kept until you implicitly remove them.

Packages That Are No Longer Required

You will probably see a message like this every time when you’re running `apt` or apt-get command (using `sudo`, of course):

The following packages were automatically installed and are no longer required:
...
Use 'sudo apt autoremove' to remove them.

This happens when you run apt/apt-get for any reason. For instance, I’m trying to install a PCAP library, but look at all the packages that can be autoremoved:

greys@xps:/home/greys$ sudo apt install libpcap-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
adobe-flashplugin  cabextract  chromium-codecs-ffmpeg-extra
 gststreamer1.0-vaapi  libaribb24-0  libavcodec-extra
 libbasicusageenvironment1  libcddb2  libdrm-dev  libdvbpsi10
 libebml4v5  libegl1-mesa-dev  libgl1-mesa-dev  libgles1
 libgles2-mesa-dev  libglu1-mesa-dev  libglvnd-core-dev  libglvnd-
 dev  libgroupsock8  libhail1-flash  libhunspell-dev
 libjs-jquery  libjs-jquery-scrollto  liblivelivemedia62  liblua5.2-0
 libmad0  libmatroska6v5  libmicrodns0  libminizip1
 libmspack0  libnfs11  libopenengl0  libopenmpt-modplug1  libplacebo4
 libprotobuf-lite10  libpthread-stubs0-dev
 libresid-builder0c2a  libSDL-image1.2  libssh2-1
 libunshield0  libupnp6  libusageenvironment3  libva-wayland2
 libvlc-bin  libvlc5  libvlccore9  libwayland-bin  libwayland-dev
 libx11-dev  libx11-xcb-dev  libxau-dev  libxcb-dri2-0-dev
 libxcb-dri3-dev  libxcb-glx0-dev  libxcb-present-dev  libxcb-
 randr0-dev  libxcb-render0-dev  libxcb-shape0-dev
 libxcb-sync-dev  libxcb-xfixes0-dev  libxcb1-dev  libxdamage-dev
 libxdmcp-dev  libxext-dev  libxfixes-dev
 libxshmfence-dev  libxxf86vm-dev  mesa-common-dev  qt5-qmake
 qt5-qmake-bin  qtmultimedia  qtx11inputplug1-unstable
 qtx11inputplug1  qttools4 qttools4-dev
 qttools4-devtools  qttools4-testing
 Need to get 221 kB of archives.
 After this operation, 748 kB of additional disk space will be used.
 Do you want to continue? [Y/n] n
 Abort.
Use apt autoremove to uninstall software packages

As suggested, this is what happens when I run apt autoremove:

greys@xps:/home/greys$ sudo apt autoremove
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
adobe-flashplugin  cabextract  chromium-codecs-ffmpeg-extra
gstreamer1.0-vaapi  libaribb24-0  libavcodec-extra
libbasicusageenvironment1  libcddb2  libdrm-dev  libdvgpu10
libebml4v5  libegl1-mesa-dev  libgl1-mesa-dev  libgles1
libgles2-mesa-dev  libglu1-mesa-dev  libgsvnd-core-dev  libgsvnd-dev
libgroupsock8  libhail1-flash  libhunspell-dev
libjs-jquery  libjs-jquery-scrollto  liblivemedia62  liblua5.2-0
libmad0  libmatroska6v5  libmicrodns0  libminizip1
libmspack0  libnfs11  libogengl0  libopenmpt-modplug1  libplacebo4
libprotobuf-lite10  libpthread-stubs0-dev
libresid-builder0c2a  libSDL-image1.2  libssh2-1
libunshield0  libupnp6  libusbagenvironment3  libva-wayland2
libvlc-bin  libvlc5  libvlc-core9  libwayland-bin  libwayland-dev
libx11-dev  libx11-xcb-dev  libxau-dev  libxcb-dri2-0-dev
libxcb-dri3-dev  libxcb-glx0-dev  libxcb-present-dev  libxcb-randr0-dev
libxcb-render0-dev  libxcb-shape0-dev
libxcb-sync-dev  libxcb-xfixes0-dev  libxcb1-dev  libx11-dev
libxdmcp-dev  libxext-dev  libxfixes-dev
libxshmfence-dev  libxxf86vm-dev  mesa-common-dev  qt5-qmake  qt5-qmake-bin qtchooser sigil-data unshield vlc-bin
vlc-data  vlc-l10n  vlc-plugin-base  vlc-plugin-notify  vlc-plugin-video-output
x11proto-core-dev  x11proto-damage-dev
x11proto-dev  x11proto-fixes-dev  x11proto-xext-dev  x11proto-xf86vidmode-dev  xorg-sgml-doctools  xtrans-dev
0 upgraded, 0 newly installed, 88 to remove and 251 not upgraded.
After this operation, 131 MB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 270228 files and directories currently installed.)
Removing adobe-flashplug (1:20190212.1-0ubuntu0.18.04.1) ...
Removing cabextract (1.6-1.1) ...
Removing chromium-codecs-ffmpeg-extra (71.0.3578.98-0ubuntu0.18.04.1) ...
Removing gstreamer1.0-vaaapi:amd64 (1.14.1-1~ubuntu18.04.1) ...
Removing vlc-plugin-base:amd64 (3.0.4-1ubuntu0.2) ...
Removing libaribb24-0:amd64 (1.0.3-1) ...
Removing libavcodec-extra (7:3.4.4-0ubuntu0.18.04.1) ...
Removing libbasicuseageenvironment1:amd64 (2018.02.18-1) ...
Removing libcdsb2 (1.3.2-5fakesync1) ...
Removing libgles2-mesa-dev:amd64 (18.2.2-0ubuntu1~18.04.2) ...
Removing libegl1-mesa-dev:amd64 (18.2.2-0ubuntu1~18.04.2) ...
Removing libgl1-mesa-dev:amd64 (9.0.0-2.1build1) ...
Removing libgl1-mesa-dev:amd64 (18.2.2-0ubuntu1~18.04.2) ...
Removing mesa-common-dev:amd64 (18.2.2-0ubuntu1~18.04.2) ...
Removing libdrm-dev:amd64 (2.4.95-1-1ubuntu0.2) ...
Removing libdvbpsi10:amd64 (1.3.2-1) ...
Removing libmatroska6v5:amd64 (1.4.8-1.1) ...
Removing libebml4v5:amd64 (1.3.5-2) ...
Removing libglvnd-dev:amd64 (1.0.0-2ubuntu2.2) ...
Removing libgles1:amd64 (1.0.0-2ubuntu2.2) ...
Removing libglvnd-core-dev:amd64 (1.0.0-2ubuntu2.2) ...
Removing libgroupsock8:amd64 (2018.02.18-1) ...
Removing libhal1-flash (0.3.3-2) ...
Removing libhunspell-dev:amd64 (1.6.2-1) ...
Removing libjs-jquery-scrollto (2.1.2+dfsg-4) ...
Removing libjs-jquery (3.2.1-1) ...
Removing liblivemedia62:amd64 (2018.02.18-1) ...
Removing liblua5.2-0:amd64 (5.2.4-1.1build1) ...
Removing libmadv0:amd64 (0.15.1b-9ubuntu18.04.1) ...
Removing libmicrodns0:amd64 (0.0.8-1) ...
Removing libminizip1:amd64 (1.1-8build1) ...
Removing libmspack0:amd64 (0.6-3ubuntu0.2) ...
Removing libnfs11:amd64 (2.0.0-1~exp1) ...
Removing libopengl0:amd64 (1.0.0-2ubuntu2.2) ...
Removing libopenmpt-modplug1:amd64 (0.3.6-1) ...
Removing vlc-plugin-video-output:amd64 (3.0.4-1ubuntu0.2) ...
Removing libplacebo4:amd64 (0.4.0-2) ...
Removing libprotobuf-lite10:amd64 (3.0.0-9.1ubuntu1) ...
Removing libx11-xcb-dev:amd64 (2:1.6.4-3ubuntu0.2) ...
Removing libxxf86vm-dev:amd64 (1:1.1.4-1) ...
Removing libresid-builder0c2a (2.1.1-15ubuntu1) ...
Removing libSDL-image1.2:amd64 (1.2.12-8) ...
Removing libsidplay2 (2.1.1-15ubuntu1) ...
Removing libssh2-1:amd64 (1.8.0-1) ...
Removing unshield (1.4.2-1) ...
Removing libunshield0:amd64 (1.4.2-1) ...
Removing libupnp6:amd64 (1:1.6.24-4) ...
Removing libusageenvironment3:amd64 (2018.02.18-1) ...
Removing libva-wayland2:amd64 (2.1.0-3) ...
Removing vlc-bin (3.0.4-1ubuntu0.2) ...
Removing libvlc-bin:amd64 (3.0.4-1ubuntu0.2) ...
Removing libvlc5:amd64 (3.0.4-1ubuntu0.2) ...
Removing vlc-plugin-notify:amd64 (3.0.4-1ubuntu0.2) ...
Removing libvlccore9:amd64 (3.0.4-1ubuntu0.2) ...
Removing libwayland-dev:amd64 (1.16.0-1ubuntu1.1~18.04.1) ...
Removing libwayland-bin (1.16.0-1ubuntu1.1~18.04.1) ...
Removing libxcb-dri2-0-dev:amd64 (1.13-1) ...
Removing libxcb-dri3-dev:amd64 (1.13-1) ...
Removing libxcb-glx0-dev:amd64 (1.13-1) ...
Removing libxcb-present-dev:amd64 (1.13-1) ...
Removing libxcb-randr0-dev:amd64 (1.13-1) ...
Removing libxcb-xfixes0-dev:amd64 (1.13-1) ...
Removing libxcb-render0-dev:amd64 (1.13-1) ...
Removing libxcb-shape0-dev:amd64 (1.13-1) ...
Removing libxcb-sync-dev:amd64 (1.13-1) ...
Removing libxdamagedev:amd64 (1:1.1.4-3) ...
Removing libxext-dev:amd64 (2:1.3.3-1) ...
Removing libxfixes-dev:amd64 (1:5.0.3-1) ...
Removing libxshmfencedev:amd64 (1.3-1) ...
Removing qt5-qmake:amd64 (5.9.5+dfsg-0ubuntu1) ...
Removing qt5-qmake-bin (5.9.5+dfsg-0ubuntu1) ...
Removing qtchooser (64-ga1b6736-5) ...
Removing sigil-data (0.9.12+dfsg-1ubuntu1804) ...
Removing vlc-data (3.0.4-1ubuntu0.2) ...
Removing vlc-l10n (3.0.4-1ubuntu0.2) ...
Removing x11proto-damage-dev (1:2018.4-4) ...
Removing x11proto-xf86vidmode-dev (2018.4-4) ...
Removing x11proto-xext-dev (2018.4-4) ...
Removing xproto-fixes-dev (1:2018.4-4) ...
Removing libx11-dev:amd64 (2:1.6.4-3ubuntu0.2) ...
Removing libxcb1-dev:amd64 (1.13-1) ...
Removing libpthread-stubs0-dev:amd64 (0.3-4) ...
Removing libxau-dev:amd64 (1:1.0.8-1) ...
Removing libxdmcp-dev:amd64 (1:1.1.2-3) ...
Removing x11proto-core-dev (2018.4-4) ...
Removing x11proto-dev (2018.4-4) ...
Removing xorg-sgml-doctools (1:1.11-1) ...
Removing xtrans-dev (1.3.5-1) ...
Processing triggers for sgml-base (1.29) ...
Processing triggers for mintsystem (8.4.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
greys@xps:/dist/wireshark/wireshark-ninja$

Just to finish this brief example, here’s what happens when I try the same apt install command again. Note how there are no more packages suggested for auto-remove:

greys@xps:/home/greys$ sudo apt install libpcap-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libpcap0.8-dev
The following NEW packages will be installed:
  libpcap-dev libpcap0.8-dev
0 upgraded, 2 newly installed, 0 to remove and 251 not upgraded.
Need to get 221 kB of archives.
After this operation, 748 kB of additional disk space will be used.
Do you want to continue? [Y/n]

See Also

- [apt](#)
- [dpkg package manager](#)
- [How To: clean apt-get cache](#)
Ubuntu SSH: How To Enable Secure Shell in Ubuntu

SSH (Secure SHell) is possibly the best way to remotely access a Unix system – it’s very secure thanks to automatic encryption of all the traffic, and it’s also quite universal because you can do all sorts of things: access remote command line shell, forward graphics session output, establish network tunnels, set up port redirections and even transfer files over the encrypted session.

Today I’m going to show you how to get started with SSH in Ubuntu.

Installing SSH server in Ubuntu

By default, your (desktop) system will have no SSH service enabled, which means you won’t be able to connect to it remotely using SSH protocol (TCP port 22). This makes installing SSH server one of the first post-install steps on your brand new Ubuntu.

The most common SSH implementation is OpenSSH. Although there are alternative implementations (closed source solutions and binary distributions maintained by various Unix and Unix-like OS vendors), OpenSSH is a de-facto standard in the secure transfers and connections industry. That’s exactly what you want to install.

Log in with your standard username and password, and run the following command to install openssh-server.

You should be using the same username that you specified when installing Ubuntu, as it will be the only account with sudo
privileges to run commands as root:

```
ubuntu$ sudo apt-get install openssh-server
[sudo] password for greys:
```

Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  rssh molly-guard openssh-blacklist openssh-blacklist-extra
The following NEW packages will be installed:
  openssh-server0 upgraded, 1 newly installed, 0 to remove and
  75 not upgraded.
Need to get 285kB of archives.
After this operation, 782kB of additional disk space will be
used.
Get:1 http://ie.archive.ubuntu.com jaunty/main openssh-server
  1:5.1p1-5ubuntu1 [285kB]
Fetched 285kB in 0s (345kB/s)
Preconfiguring packages ...
Selecting previously deselected package openssh-server.
(Reading database ... 101998 files and directories currently
installed.)
Unpacking openssh-server (from .../openssh-server_1%3a5.1p1-5ubuntu1_i386.deb) ...
Processing triggers for ufw ...
Processing triggers for man-db ...
Setting up openssh-server (1:5.1p1-5ubuntu1) ...
Creating SSH2 RSA key; this may take some time ...
Creating SSH2 DSA key; this may take some time ...
* Restarting OpenBSD Secure Shell server sshd [ OK ]

**Public and Private keys in SSH**

As you can see in the sample output above, the installation
procedure created 2 sets of keys – SSH2 RSA keypair and SSH2
DSA keypair. The reason for this is that OpenSSH relies
heavily on the public and private key (PPK) infrastructure.

The concept behind PPK is pretty cool: SSH allows you to
create keypairs. They are generated to the maximum randomness
achievable on your system. Keypairs can be created for your server or for your individual uses.

The idea is that public keys are shared with other servers, and they later can be used as a unique identificator to confirm your true identity. When you’re connecting to another server, it uses your public key to encrypt a short message and the secure session will only be established if on your side you have a private key that allows decrypting the message. No other system or user can decrypt the message because only you would have the private key. That’s why it’s called private – don’t ever share it with anyone.

As an additional security measure, when you’re generating personal keypairs you’ll be asked to supply a passphrase so that even if someone steals your private password they won’t be able to use it without knowing your passphrase.

**Verifying your SSH server works**

While you’re still on your local desktop session, you can use the `ps` command to confirm that SSH daemon (`sshd`) is running:

```
ubuntu$ ps -aef | grep sshd
root    24114     1  0 15:18 ?        00:00:00 /usr/sbin/sshd
```

Now that you see it’s there, it’s time to try connecting:

```
ubuntu$ ssh localhost
```

Since this is the first time you’re trying to connect using SSH, you’ll have to answer yes to the following question:

The authenticity of host 'localhost (::1)' can't be established. RSA key fingerprint is 18:4d:96:b3:0d:25:00:c8:a1:a3:84:5c:9f:1c:0d:a5. Are you sure you want to continue connecting (yes/no)? yes

… you’ll then be prompted for your own password (remember, the system treats such connection request as if you were connecting remotely, so it can’t trust you without confirming
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.

.. and finally you’ll see the usual Ubuntu (Jaunty in this example) banner and prompt:

Linux ubuntu 2.6.28-11-generic #42-Ubuntu SMP Fri Apr 17 01:57:59 UTC 2009 i686

The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

To access official Ubuntu documentation, please visit: http://help.ubuntu.com

Last login: Fri May 15 15:18:34 2009 from ubuntu

ubuntu$

That’s it, providing you have your networking configured and you know your IP address or hostname, you can start connecting to your Ubuntu box from remote systems, using the same command. Enjoy!

Recommended books:

See also:

  - Passwordless SSH
  - Unix commands
- Using apt-get behind proxy
- List installed packages in Ubuntu