



## CONTENTS

Package Management Systems: A Brief Overview

Update Package Lists

Upgrade Installed Packages

Find a Package

View Info About a Specific Package

Install a Package from Repositories

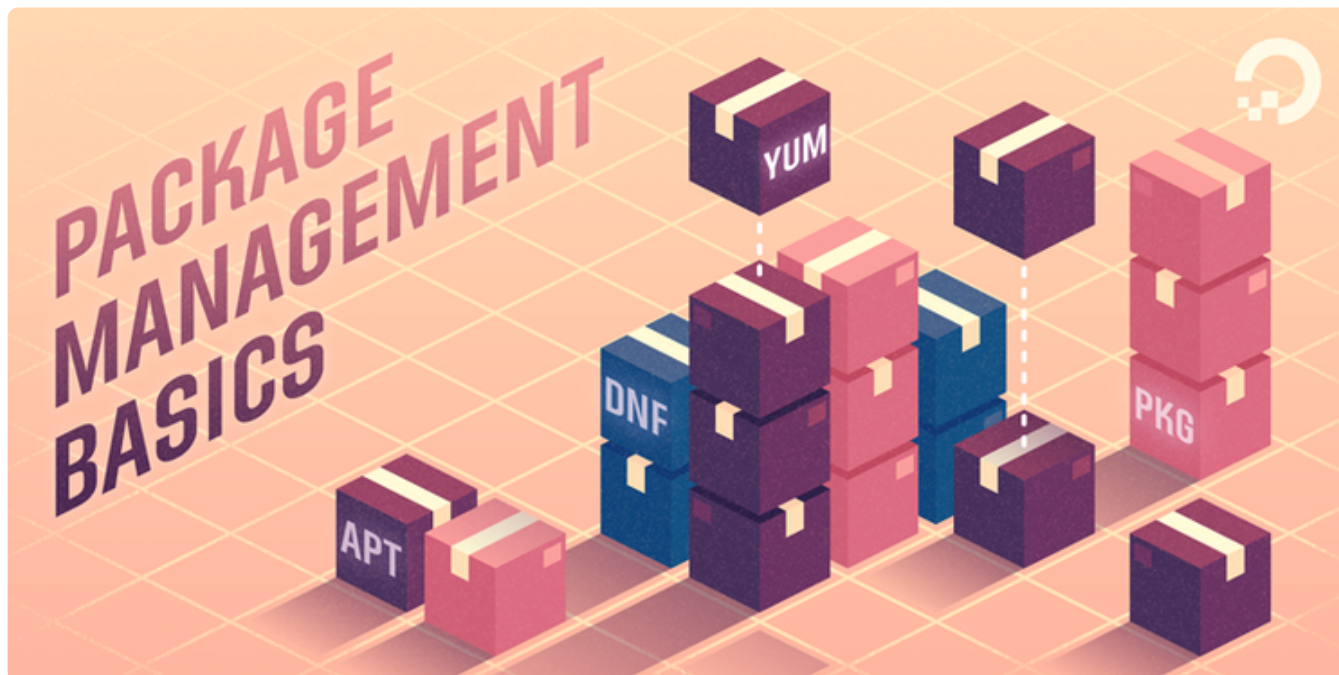
Install a Package from the Local Filesystem

Remove One or More Installed Packages

The apt Command

Get Help

Conclusion and Further Reading



## RELATED

How To Set Up a Remote Desktop with X2Go on Ubuntu 20.04

[Tutorial](#)

How To Centralize Logs With JournalD on Ubuntu 20.04

[Tutorial](#)

## TUTORIAL

# Package Management Basics: apt, yum, dnf, pkg

Linux Basics

Getting Started

System Tools

Linux Commands



By [Brennen Bearnes](#)

Posted January 4, 2016 400.9k

## Introduction

Most modern Unix-like operating systems offer a centralized mechanism for finding and installing software. Software is usually distributed in the form of **packages**, kept in **repositories**. Working with packages is known as **package management**. Packages provide the basic components of an operating system, along with shared libraries, applications, services, and documentation.

A package management system does much more than one-time installation of software. It also provides tools for upgrading already-installed packages. Package repositories help to ensure that code has been vetted for use on your system, and that the installed versions of software have been approved by developers and package maintainers.

When configuring servers or development environments, it's often necessary look beyond official repositories. Packages in the stable release of a distribution may be out of date, especially where new or rapidly-changing software is concerned. Nevertheless, package management is a vital skill for system administrators and developers, and the wealth of packaged software for major distributions is a tremendous resource.

This guide is intended as a quick reference for the fundamentals of finding, installing, and upgrading packages on a variety of distributions, and should help you translate that knowledge between systems.

## Package Management Systems: A Brief Overview

Most package systems are built around collections of package files. A package file is usually an archive which contains compiled binaries and other resources making up the software, along with installation scripts. Packages also contain valuable metadata, including their **dependencies**, a list of other packages required to install and run them.

While their functionality and benefits are broadly similar, packaging formats and tools vary by platform:

Operating System	Format	Tool(s)

Debian	.deb	apt, apt-cache, apt-get, dpkg
Ubuntu	.deb	apt, apt-cache, apt-get, dpkg
CentOS	.rpm	yum
Fedora	.rpm	dnf
FreeBSD	Ports, .txz	make, pkg

In Debian and systems based on it, like Ubuntu, Linux Mint, and Raspbian, the package format is the `.deb` file. APT, the Advanced Packaging Tool, provides commands used for most common operations: Searching repositories, installing collections of packages and their dependencies, and managing upgrades. APT commands operate as a front-end to the lower-level `dpkg` utility, which handles the installation of individual `.deb` files on the local system, and is sometimes invoked directly.

Recent releases of most Debian-derived distributions include the `apt` command, which offers a concise and unified interface to common operations that have traditionally been handled by the more-specific `apt-get` and `apt-cache`. Its use is optional, but may simplify some tasks.

CentOS, Fedora, and other members of the Red Hat family use RPM files. In CentOS, `yum` is used to interact with both individual package files and repositories.

In recent versions of Fedora, `yum` has been supplanted by `dnf`, a modernized fork which retains most of `yum`'s interface.

FreeBSD's binary package system is administered with the `pkg` command. FreeBSD also offers the Ports Collection, a local directory structure and tools which allow the user to fetch, compile, and install packages directly from source using Makefiles. It's usually much more convenient to use `pkg`, but occasionally a pre-compiled package is unavailable, or you may need to change compile-time options.

## Update Package Lists

Most systems keep a local database of the packages available from remote repositories. It's best to update this database before installing or upgrading packages. As a partial exception to this pattern, `yum` and `dnf` will check for updates before performing some operations, but you can ask them at any time whether updates are available.

System	Command
Debian / Ubuntu	<code>sudo apt-get update</code>
	<code>sudo apt update</code>
CentOS	<code>yum check-update</code>
Fedora	<code>dnf check-update</code>
FreeBSD Packages	<code>sudo pkg update</code>
FreeBSD Ports	<code>sudo portsnap fetch update</code>

## Upgrade Installed Packages

Making sure that all of the installed software on a machine stays up to date would be an enormous undertaking without a package system. You would have to track upstream changes and security alerts for hundreds of different packages. While a package manager doesn't solve every problem you'll encounter when upgrading software, it does enable you to maintain most system components with a few commands.

On FreeBSD, upgrading installed ports can introduce breaking changes or require manual configuration steps. It's best to read `/usr/ports/UPDATING` before upgrading with `portmaster`.

System	Command	Notes
Debian /	<code>sudo apt-get upgrade</code>	Only upgrades installed

Ubuntu		packages, where possible.
	<code>sudo apt-get dist-upgrade</code>	May add or remove packages to satisfy new dependencies.
	<code>sudo apt upgrade</code>	Like <code>apt-get upgrade</code> .
	<code>sudo apt full-upgrade</code>	Like <code>apt-get dist-upgrade</code> .
CentOS	<code>sudo yum update</code>	
Fedora	<code>sudo dnf upgrade</code>	
FreeBSD Packages	<code>sudo pkg upgrade</code>	
FreeBSD Ports	<code>less /usr/ports/UPDATING</code>	Uses <code>less</code> to view update notes for ports (use arrow keys to scroll, press <b>q</b> to quit).
	<code>cd /usr/ports/ports-mgmt/portmaster &amp;&amp; sudo make install &amp;&amp; sudo portmaster -a</code>	Installs <code>portmaster</code> and uses it to update installed ports.

## Find a Package

Most distributions offer a graphical or menu-driven front end to package collections. These can be a good way to browse by category and discover new software. Often, however, the quickest and most effective way to locate a package is to search with command-line tools.

System	Command	Notes

Debian / Ubuntu	<code>apt-cache search search_string</code>	
	<code>apt search search_string</code>	
CentOS	<code>yum search search_string</code>	
	<code>yum search all search_string</code>	Searches all fields, including description.
Fedora	<code>dnf search search_string</code>	
	<code>dnf search all search_string</code>	Searches all fields, including description.
FreeBSD Packages	<code>pkg search search_string</code>	Searches by name.
	<code>pkg search -f search_string</code>	Searches by name, returning full descriptions.
	<code>pkg search -D search_string</code>	Searches description.
FreeBSD Ports	<code>cd /usr/ports &amp;&amp; make search name=package</code>	Searches by name.
	<code>cd /usr/ports &amp;&amp; make search key=search_string</code>	Searches comments, descriptions, and dependencies.

## View Info About a Specific Package

When deciding what to install, it's often helpful to read detailed descriptions of packages. Along with human-readable text, these often include metadata like version numbers and a list of the package's dependencies.

System	Command	Notes
Debian / Ubuntu	<code>apt-cache show package</code>	Shows locally-cached info about a package.
	<code>apt show package</code>	
	<code>dpkg -s package</code>	Shows the current installed status of a package.
CentOS	<code>yum info package</code>	
	<code>yum deplist package</code>	Lists dependencies for a package.
Fedora	<code>dnf info package</code>	
	<code>dnf repoquery --requires package</code>	Lists dependencies for a package.
FreeBSD Packages	<code>pkg info package</code>	Shows info for an installed package.
FreeBSD Ports	<code>cd /usr/ports/category/port &amp;&amp; cat pkg-descr</code>	

## Install a Package from Repositories

Once you know the name of a package, you can usually install it and its dependencies with a single command. In general, you can supply multiple packages to install simply by listing them all.

System	Command	Notes
Debian / Ubuntu	<code>sudo apt-get install package</code>	
	<code>sudo apt-get install package1 package2 ...</code>	Installs all listed packages.
	<code>sudo apt-get install -y package</code>	Assumes “yes” where <code>apt</code> would usually prompt to continue.
	<code>sudo apt install package</code>	Displays a colored progress bar.
CentOS	<code>sudo yum install package</code>	
	<code>sudo yum install package1 package2 ...</code>	Installs all listed packages.
	<code>sudo yum install -y package</code>	Assumes “yes” where <code>yum</code> would usually prompt to continue.
Fedora	<code>sudo dnf install package</code>	
	<code>sudo dnf install package1 package2 ...</code>	Installs all listed packages.
	<code>sudo dnf install -y package</code>	Assumes “yes” where <code>dnf</code> would usually



		prompt to continue.
FreeBSD Packages	<code>sudo pkg install package</code>	
	<code>sudo pkg install package1 package2 ...</code>	Installs all listed packages.
FreeBSD Ports	<code>cd /usr/ports/category/port &amp;&amp; sudo make install</code>	Builds and installs a port from source.

## Install a Package from the Local Filesystem

Sometimes, even though software isn't officially packaged for a given operating system, a developer or vendor will offer package files for download. You can usually retrieve these with your web browser, or via `curl` on the command line. Once a package is on the target system, it can often be installed with a single command.

On Debian-derived systems, `dpkg` handles individual package files. If a package has unmet dependencies, `gdebi` can often be used to retrieve them from official repositories.

On CentOS and Fedora systems, `yum` and `dnf` are used to install individual files, and will also handle needed dependencies.

System	Command	Notes
Debian / Ubuntu	<code>sudo dpkg -i package.deb</code>	
	<code>sudo apt-get install -y gdebi &amp;&amp; sudo gdebi package.deb</code>	Installs and uses <code>gdebi</code> to install <code>package.deb</code>

		and retrieve any missing dependencies.
CentOS	<code>sudo yum install package.rpm</code>	
Fedora	<code>sudo dnf install package.rpm</code>	
FreeBSD Packages	<code>sudo pkg add package.txz</code>	
	<code>sudo pkg add -f package.txz</code>	Installs package even if already installed.

## Remove One or More Installed Packages

Since a package manager knows what files are provided by a given package, it can usually remove them cleanly from a system if the software is no longer needed.

System	Command	Notes
Debian / Ubuntu	<code>sudo apt-get remove package</code>	
	<code>sudo apt remove package</code>	
	<code>sudo apt-get autoremove</code>	Removes unneeded packages.
CentOS	<code>sudo yum remove package</code>	
Fedora	<code>sudo dnf erase package</code>	
FreeBSD	<code>sudo pkg delete package</code>	

Packages		
	<code>sudo pkg autoremove</code>	Removes unneeded packages.
FreeBSD Ports	<code>sudo pkg delete package</code>	
	<code>cd /usr/ports/path_to_port &amp;&amp; make deinstall</code>	De-installs an installed port.

## The `apt` Command

Administrators of Debian-family distributions are generally familiar with `apt-get` and `apt-cache`. Less widely known is the simplified `apt` interface, designed specifically for interactive use.

Traditional Command	<code>apt</code> Equivalent
<code>apt-get update</code>	<code>apt update</code>
<code>apt-get dist-upgrade</code>	<code>apt full-upgrade</code>
<code>apt-cache search string</code>	<code>apt search string</code>
<code>apt-get install package</code>	<code>apt install package</code>
<code>apt-get remove package</code>	<code>apt remove package</code>
<code>apt-get purge package</code>	<code>apt purge package</code>

While `apt` is often a quicker shorthand for a given operation, it's not intended as a complete replacement for the traditional tools, and its interface may change between versions to improve usability. If you are using package management commands inside a script or a shell pipeline, it's a good idea to stick with `apt-get` and `apt-cache`.

# Get Help

In addition to web-based documentation, keep in mind that Unix manual pages (usually referred to as **man pages**) are available for most commands from the shell. To read a page, use `man`:

```
$ man page
```

In `man`, you can navigate with the arrow keys. Press `/` to search for text within the page, and `q` to quit.

System	Command	Notes
Debian / Ubuntu	<code>man apt-get</code>	Updating the local package database and working with packages.
	<code>man apt-cache</code>	Querying the local package database.
	<code>man dpkg</code>	Working with individual package files and querying installed packages.
	<code>man apt</code>	Working with a more concise, user-friendly interface to most basic operations.
CentOS	<code>man yum</code>	
Fedora	<code>man dnf</code>	
FreeBSD Packages	<code>man pkg</code>	Working with pre-compiled binary packages.
FreeBSD Ports	<code>man ports</code>	Working with the Ports Collection.

## Conclusion and Further Reading

This guide provides an overview of basic operations that can be cross-referenced between systems, but only scratches the surface of a complex topic. For greater detail on a given system, you can consult the following resources:

- [This guide](#) covers Ubuntu and Debian package management in detail.
- There's an [official CentOS guide to managing software with yum](#).
- There's a [Fedora wiki page about dnf](#), and an [official manual for dnf](#) itself.
- [This guide](#) covers FreeBSD package management using `pkg`.
- The [FreeBSD Handbook](#) contains a [section on using the Ports Collection](#).

Was this helpful?

Yes

No



[Report an issue](#)

## About the authors



**[Brennen Bearnes](#)**

has authored 48 tutorials.

## Related

## TUTORIAL

### How To Set Up a Remote Desktop with X2Go on Ubuntu 20.04

X2Go provides a working "cloud desktop," complete with all the advantages of an always-online, ...

## TUTORIAL

### How To Centralize Logs With Journald on Ubuntu 20.04

In this guide, you will configure a component of the systemd suite of tools to relay log messages ...

## TUTORIAL

### How To Set Up a Remote Desktop with X2Go on Debian 10

X2Go provides a working "cloud desktop," complete with all the advantages of an always-online, ...

## TUTORIAL

### How To Upgrade to Ubuntu 20.04 Focal Fossa

The Ubuntu operating system's latest Long Term Support (LTS) release, Ubuntu 20.04 (Focal ...

## Still looking for an answer?



Ask a question



Search for more help

## Comments

### 6 Comments

**B** *I*



Leave a comment...

Sign In to Comment



**matze** January 25, 2016

3

you can use just “apt” for everything (except dpkg)

apt-get update -> apt update

apt-get upgrade -> apt upgrade

apt-get dist-upgrade -> apt dist-upgrade

apt-cache search *searchstring* -> *apt search searchstring*

apt-cache show package -> apt show package

install

remove

...

[Reply](#) [Report](#)



**samob** March 31, 2016

0

Great article! Apt-ollutly fantastic, condensed and straightforward :) Just wondering, did Fedora ppl ever get away from computers? I mean, in sports dnf means “did not finish”....

[Reply](#) [Report](#)



**olegkorol91** March 31, 2016

1

What about the best package managers, pacman and zypper?

You forgot us, the Arch users... ;)

PS: Good article :)

[Reply](#) [Report](#)



**bpb** March 31, 2016

To be completely transparent about this, we decided to document distributions that we support right now on DigitalOcean. Partly to keep the guide to a reasonable length, but mostly because we don't want to confuse users about what's on our platform specifically.

We don't mean to make any kind of statement with the exclusion of other systems. I'm an occasional Arch user myself, and I must land on ArchWiki once a week or better while researching one topic or another. :)

[Reply](#) [Report](#)



**olegkorol91** March 31, 2016

Totally understand your point.

You are doing a great job, guys, keep it up.

Glad to be a DO-user. In my opinion, the best VPS provider out there (with a great community).

[Reply](#) [Report](#)



**negiveer** July 27, 2018

helpful but i m want become Linux administrator how it possible

[Reply](#) [Report](#)



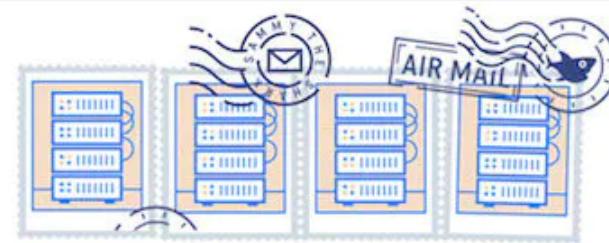
This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.





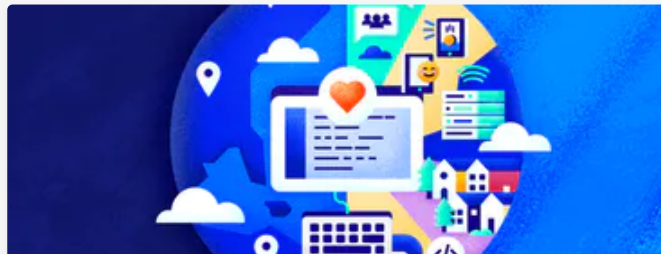
#### **BECOME A CONTRIBUTOR**

You get paid; we donate to tech nonprofits.



#### **GET OUR BIWEEKLY NEWSLETTER**

Sign up for Infrastructure as a Newsletter.



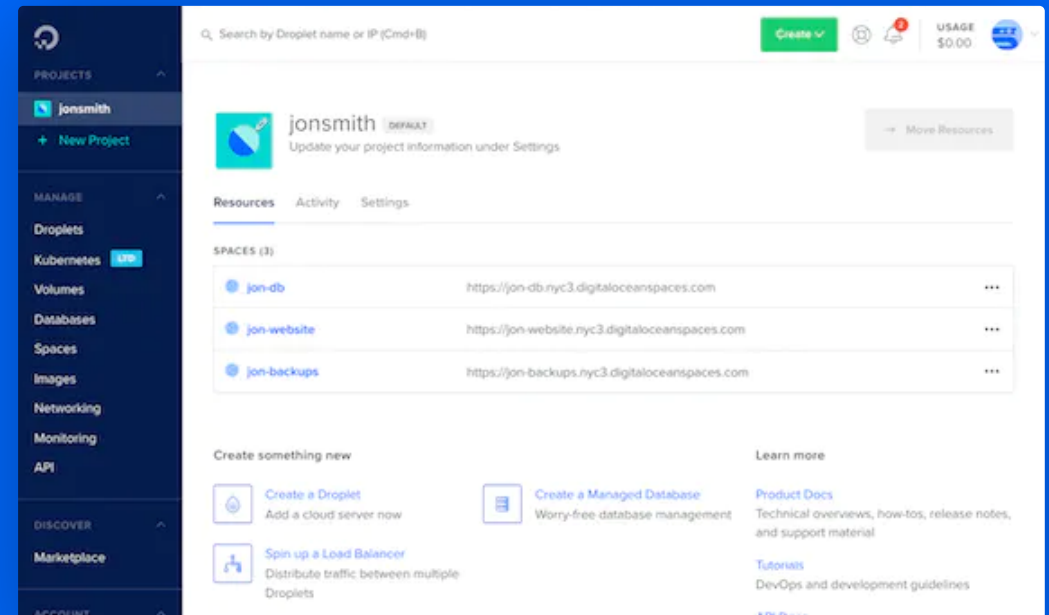
#### **HUB FOR GOOD**

Working on improving health and education, reducing inequality, and spurring economic growth? We'd like to help.

# Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

[Learn More](#)



© 2020 DigitalOcean, LLC. All rights reserved.

## Company

[About](#)  
[Leadership](#)  
[Blog](#)  
[Careers](#)  
[Partners](#)  
[Referral Program](#)  
[Press](#)  
[Legal](#)  
[Security & Trust Center](#)

## Products

[Pricing](#)  
[Products Overview](#)  
[Droplets](#)  
[Kubernetes](#)  
[Managed Databases](#)  
[Spaces](#)  
[Marketplace](#)  
[Load Balancers](#)  
[Block Storage](#)  
[API Documentation](#)

## Community

[Tutorials](#)  
[Q&A](#)  
[Tools and Integrations](#)  
[Tags](#)  
[Product Ideas](#)  
[Write for DigitalOcean](#)  
[Presentation Grants](#)  
[Hatch Startup Program](#)  
[Shop Swag](#)  
[Research Program](#)

## Contact

[Get Support](#)  
[Trouble Signing In?](#)  
[Sales](#)  
[Report Abuse](#)  
[System Status](#)

We use cookies to provide our services and for analytics and marketing. To find out more about our use of cookies, please see our [Privacy Policy](#). By continuing to browse our website, you agree to our use of cookies.

OK