Third Edition

The Official ubuntu Server Book

Kyle Rankin
Benjamin Mako Hill

Praise for *The Official Ubuntu Server Book*

"Murphy's Law is never truer than when it comes to administering a Linux server. You can pretty much count on something happening to your machine at a time when you need it the most. That's when a book with some basic troubleshooting instructions is worth every penny you paid for it. Chapter 11 covers the steps you should take when something goes wrong."

—Paul Ferrill, LinuxPlanet.com reviewer

"College-level collections catering to Linux programmers and developers will find *The Official Ubuntu Server Book,* a top addition to the collection, covering a complete, free server operating system in a guide to getting going quickly. From making the most of Ubuntu Server's latest technologies to automating installs and protecting the server using Ubuntu's built-in security tools, *The Official Ubuntu Server Book,* is packed with keys to success for any Ubuntu user."

—Jim Cox, *Midwest Book Review*

"This book will get you started on the path of the server admin, both within the context of Ubuntu server and in the larger realm of managing a server infrastructure. The desktop and server versions of Ubuntu are continuing to mature. Read this book if you want to keep up."

—James Pyles, author

CHAPTER 5

Guide to Common Ubuntu Servers

WHILE LINUX HAS BEEN USED AS a desktop operating system for a long time, it has arguably been used longer and by many more people as a server. Over the years Linux has accumulated hundreds of different services either ported from another operating system or developed primarily on Linux. This means that when you want to use Ubuntu as a server, there are literally hundreds of different servers Ubuntu could be, many of which can be installed by a simple call to a package manager.

It would be extremely difficult, if not impossible, to document all of Ubuntu's possible services in a book, much less a chapter. What I realize, though, is that among the hundreds of services you could install, there are a handful of common services that most administrators deal with on a daily basis. This chapter covers some of the most common Ubuntu server types that an administrator will run into. If you are a beginning administrator, I will provide you with a step-by-step guide to set up what might be your first DNS server with some common configuration schemes and best practices. If you are an advanced administrator, think of this chapter as a guide to the Ubuntu approach to servers. I show you how Ubuntu organizes configuration files along with any particular tools or shortcuts it provides for a service. I also cover major file locations so you'll know exactly where to look for the core configuration files, which init scripts are important, and where logs are stored.

Entire books have been written about how to administer basically every service discussed here. I don't attempt to document all of the major Apache modules, for instance, nor do I discuss every configuration option for BIND or Postfix. What I give you is a good departure point with working example configuration files. A sysadmin is always busy, so where Ubuntu has provided time-saving shortcuts, I point them out. After all, you only have to do something the hard, "old-school," time-consuming way so many times before it loses its mystique.

DNS Server

The first service I discuss also happens to be one of the oldest. DNS, or Domain Name System, is one of the fundamental services that keep the Internet running. In a nutshell, DNS is the service that among other things translates a hostname, such as www.ubuntu.com, into an IP address, like

91.189.94.8. DNS servers maintain a local directory of names and IP addresses for which they are authoritative, so that if you ask a DNS server for the IP address for a name for which it is authoritative, it should generally respond with the answer. Many DNS servers can serve recursive requests as well. With a recursive query, a DNS server might not itself contain the record you are looking for, but it can go out on the Internet and find the answer for you.

There are many different programs that provide DNS services on Ubuntu, but the most common has also been around the longest—BIND. BIND, short for Berkeley Internet Name Domain, has many advanced features, but with many features often comes a more complicated configuration file. Honestly, what trips up most administrators is simply building a BIND server from scratch, but as you will see, under Ubuntu the heavy lifting has already been done for you.

Install BIND

There are a few ways to install BIND under Ubuntu. During the initial Ubuntu install you can select DNS Server from the list of server types. If you have already installed Ubuntu, you can type `sudo tasksel` to get to the same menu. Once you select DNS Server and then OK, the bind9 and bind9-doc packages will automatically be downloaded and installed for you. Alternatively you can just run

```
$ sudo apt-get install bind9 bind9-doc
```

What you will discover (and what I cover in more detail next) is that once the Ubuntu BIND package has been installed, you will actually have a fully configured and functioning name server ready, at least, to act as a caching name server for recursive queries. Sample configuration files, root zone files, local zones, and even `rndc` keys have already been configured for you.

Ubuntu Conventions

There are a few key file conventions that Ubuntu's BIND uses that might be different from what you are accustomed to, depending on what other

Linux distributions you have used. Here are some of the key Ubuntu file locations:

- **/etc/bind/**
  This directory contains the main BIND configuration file, named.conf, as well as any individual zone files. Any new master zone files should also be stored in this directory or, if you have many zone files you wish to organize, in a subdirectory below /etc/bind.

- **/etc/bind/named.conf**
  This is BIND's main configuration file and is where you change BIND's options and behavior. All of BIND's individual zone files (files containing name and IP information) are referenced in /etc/bind/named.conf.local. The main options that are enabled in BIND are included from a separate file named /etc/bind/named.conf.options.

- **/etc/bind/db.***
  As a convention, all zone files start with db. and then some name or number to identify the particular zone. Names are typically used when the zone contains traditional forward DNS records (names mapped to IPs), and numbers are typically used for reverse DNS records (IPs mapped back to a name). For instance, by default Ubuntu's BIND will include a few zone files such as db.root (information about the root name servers on the Internet), db.local (localhost zone information), and db.127 (reverse DNS records for localhost).

- **/var/cache/bind**
  This is BIND's working directory and where it will store slave zone files. If your server will act as a slave for a particular zone, configure it to store its files here.

- **/etc/init.d/bind9**
  This is BIND's init script. Once you install the `bind9` package, it will automatically be set up to start on system boot, but you can run the init script manually with /etc/init.d/bind9, or run `sudo` service `bind9`.

- **/var/log/syslog**
  This is the default log file for BIND. A number of different services log to this file, but log entries for BIND will be prefixed by the keyword `named`, so if you wanted to see only the BIND log entries you could run `sudo grep named /var/log/syslog`.

Caching Name Server

The default Ubuntu BIND configuration is ready out of the box to be a caching name server. Essentially a caching name server acts as a middleman for DNS queries. Once you configure some hosts to point to a caching name server, when one host requests a particular record, the caching name server goes out to the Internet, retrieves a record, and stores that record locally. If a second host requests the same record, and that record hasn't expired from the cache, a caching name server will simply return the cached result. This can dramatically improve the response times for a network of hosts, especially for Web browsing, since often the same records (like, say, www.google.com) are requested by multiple hosts.

DNS Master

Caching name servers are very useful, but when most people install BIND, they intend to host some zone files of their own. When a name server hosts zone files locally and doesn't need to retrieve them from any other source, it is known as a master. When you want to add zones to a DNS server, there are basically three steps: Create a zone file, add a reference to that zone file in named.conf.local, and tell BIND to reload its configuration.

For my example let's assume that I have a name server inside my network at 192.168.0.5 and I registered example.net. I want this name server to have the following entries:

- ns1.example.net points to 192.168.0.5 (the name server itself).
- example.net also points to 192.168.0.7.
- www.example.net points to 192.168.0.7.
- gateway.example.net points to 192.168.0.1.

The simplest way to create a new zone file is to copy one you already have and change it. In this case the best candidate is the /etc/bind/ db.local file, so I copy it to db.example.net:

```
$ sudo cp /etc/bind/db.local /etc/bind/db.example.net
```

When I open db.example.net in a text editor, I will see the following configuration:

```
;
; BIND data file for local loopback interface
;
$TTL   604800
@   IN   SOA   localhost. root.localhost. (
                     2         ; Serial
                604800         ; Refresh
                 86400         ; Retry
                2419200         ; Expire
                 604800 )      ; Negative Cache TTL
;
@    IN    NS   localhost.
@    IN    A    127.0.0.1
@    IN    AAAA ::1
```

Because this isn't intended to be a complete guide to BIND I won't go into every option in this file, but by default it is configured for a TTL (Time to Live, the amount of time before a name server that has requested a record considers it stale) of 604,800 seconds, or seven days. Next it lists localhost as the SOA (Start of Authority, the server that should be considered the best source of information), and root@localhost is the contact e-mail address to use for this host (referenced by root.localhost). Further down, the file lists localhost as a name server for this zone with an NS record, then sets localhost's IP address to be 127.0.0.1 and even adds an IPv6 address for localhost with the AAAA record.

I then changed this record to suit the requirements I set out previously, and this is the resulting zone file:

```
;
; BIND data file for example.net
;
$TTL   1d
```