DigitalOcean

Subscribe Share Contents

95

How To Install and Configure Redis on Ubuntu 16.04

Posted May 11, 2016 520.3k NOSQL REDIS UBUNTU UBUNTU 16.04

By: Justin Ellingwood

Not using **Ubuntu 16.04**? Choose a different version:

Introduction

Redis is an in-memory key-value store known for its flexibility, performance, and wide language support. In this guide, we will demonstrate how to install and configure Redis on an Ubuntu 16.04 server.

Prerequisites

To complete this guide, you will need access to an Ubuntu 16.04 server. You will need a non-root user with `sudo` privileges to perform the administrative functions required for this process. You can learn how to set up an account with these privileges by following our Ubuntu 16.04 initial server setup guide.

When you are ready to begin, log in to your Ubuntu 16.04 server with your `sudo` user and continue below.

Install the Build and Test Dependencies

In order to get the latest version of Redis, we will be compiling and installing the software from source. Before we download the code, we need to satisfy the build dependencies so that we can compile the software.

To do this, we can install the `build-essential` meta-package from the Ubuntu repositories. We will also be downloading the `tcl` package, which we can use to test our binaries.

We can update our local `apt` package cache and install the dependencies by typing:

```
$ sudo apt-get update
$ sudo apt-get install build-essential tcl
```

Download, Compile, and Install Redis

Next, we can begin to build Redis.

Download and Extract the Source Code

Since we won't need to keep the source code that we'll compile long term (we can always re-download it), we will build in the `/tmp` directory. Let's move there now:

```
$ cd /tmp
```

Now, download the latest stable version of Redis. This is always available at a stable download URL:

```
$ curl -O http://download.redis.io/redis-stable.tar.gz
```

Unpack the tarball by typing:

```
$ tar xzvf redis-stable.tar.gz
```

Move into the Redis source directory structure that was just extracted:

```
$ cd redis-stable
```

Build and Install Redis

Now, we can compile the Redis binaries by typing:

```
$ make
```

After the binaries are compiled, run the test suite to make sure everything was built correctly. You can do this by typing:

```
$ make test
```

This will typically take a few minutes to run. Once it is complete, you can install the binaries onto the system by typing:

```
$ sudo make install
```

Configure Redis

Now that Redis is installed, we can begin to configure it.

To start off, we need to create a configuration directory. We will use the conventional `/etc/redis` directory, which can be created by typing:

```
$ sudo mkdir /etc/redis
```

Now, copy over the sample Redis configuration file included in the Redis source archive:

```
$ sudo cp /tmp/redis-stable/redis.conf /etc/redis
```

Next, we can open the file to adjust a few items in the configuration:

```
$ sudo nano /etc/redis/redis.conf
```

In the file, find the `supervised` directive. Currently, this is set to `no`. Since we are running an operating system that uses the systemd init system, we can change this to `systemd`:

/etc/redis/redis.conf

```
. . .

# If you run Redis from upstart or systemd, Redis can interact with your
# supervision tree. Options:
#   supervised no      - no supervision interaction
#   supervised upstart - signal upstart by putting Redis into SIGSTOP mode
#   supervised systemd - signal systemd by writing READY=1 to $NOTIFY_SOCKET
#   supervised auto    - detect upstart or systemd method based on
#                        UPSTART_JOB or NOTIFY_SOCKET environment variables
# Note: these supervision methods only signal "process is ready."
#       They do not enable continuous liveness pings back to your supervisor.
supervised systemd

. . .
```

Next, find the `dir` directory. This option specifies the directory that Redis will use to dump persistent data. We need to pick a location that Redis will have write permission and that isn't viewable by normal users.

We will use the `/var/lib/redis` directory for this, which we will create in a moment:

/etc/redis/redis.conf

```
. . .

# The working directory.
#
# The DB will be written inside this directory, with the filename specified
# above using the 'dbfilename' configuration directive.
#
# The Append Only File will also be created inside this directory.
#
# Note that you must specify a directory here, not a file name.
dir /var/lib/redis

. . .
```

Save and close the file when you are finished.

Create a Redis systemd Unit File

Next, we can create a systemd unit file so that the init system can manage the Redis process.

Create and open the `/etc/systemd/system/redis.service` file to get started:

```
$ sudo nano /etc/systemd/system/redis.service
```

Inside, we can begin the `[Unit]` section by adding a description and defining a requirement that networking be available before starting this service:

/etc/systemd/system/redis.service

```
[Unit]
Description=Redis In-Memory Data Store
After=network.target
```

In the `[Service]` section, we need to specify the service's behavior. For security purposes, we should not run our service as `root`. We should use a dedicated user and group, which we will call `redis` for simplicity. We will create these momentarily.

To start the service, we just need to call the `redis-server` binary, pointed at our configuration. To stop it, we can use the Redis `shutdown` command, which can be executed with the `redis-cli` binary. Also, since we want Redis to recover from failures when possible, we will set the `Restart` directive to "always":

/etc/systemd/system/redis.service

```
[Unit]
Description=Redis In-Memory Data Store
After=network.target

[Service]
User=redis
Group=redis
ExecStart=/usr/local/bin/redis-server /etc/redis/redis.conf
ExecStop=/usr/local/bin/redis-cli shutdown
Restart=always
```

Finally, in the `[Install]` section, we can define the systemd target that the service should attach to if enabled (configured to start at boot):

/etc/systemd/system/redis.service

```
[Unit]
Description=Redis In-Memory Data Store
```

```
After=network.target

[Service]
User=redis
Group=redis
ExecStart=/usr/local/bin/redis-server /etc/redis/redis.conf
ExecStop=/usr/local/bin/redis-cli shutdown
Restart=always

[Install]
WantedBy=multi-user.target
```

Save and close the file when you are finished.

Create the Redis User, Group and Directories

Now, we just have to create the user, group, and directory that we referenced in the previous two files.

Begin by creating the `redis` user and group. This can be done in a single command by typing:

```
$ sudo adduser --system --group --no-create-home redis
```

Now, we can create the `/var/lib/redis` directory by typing:

```
$ sudo mkdir /var/lib/redis
```

We should give the `redis` user and group ownership over this directory:

```
$ sudo chown redis:redis /var/lib/redis
```

Adjust the permissions so that regular users cannot access this location:

```
$ sudo chmod 770 /var/lib/redis
```

Start and Test Redis

Now, we are ready to start the Redis server.

Start the Redis Service

Start up the systemd service by typing:

```
$ sudo systemctl start redis
```

Check that the service had no errors by running:

```
$ sudo systemctl status redis
```

You should see something that looks like this:

```
Output
● redis.service - Redis Server
   Loaded: loaded (/etc/systemd/system/redis.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2016-05-11 14:38:08 EDT; 1min 43s ago
  Process: 3115 ExecStop=/usr/local/bin/redis-cli shutdown (code=exited, status=0/SUCCESS)
 Main PID: 3124 (redis-server)
    Tasks: 3 (limit: 512)
   Memory: 864.0K
      CPU: 179ms
   CGroup: /system.slice/redis.service
           └─3124 /usr/local/bin/redis-server 127.0.0.1:6379

. . .
```

Test the Redis Instance Functionality

To test that your service is functioning correctly, connect to the Redis server with the command-line client:

```
$ redis-cli
```

In the prompt that follows, test connectivity by typing:

```
127.0.0.1:6379> ping
```

You should see:

```
Output
PONG
```

Check that you can set keys by typing:

```
127.0.0.1:6379> set test "It's working!"
```

```
Output
OK
```

Now, retrieve the value by typing:

```
127.0.0.1:6379> get test
```

You should be able to retrieve the value we stored:

```
Output
"It's working!"
```

Exit the Redis prompt to get back to the shell:

```
127.0.0.1:6379> exit
```

As a final test, let's restart the Redis instance:

```
$ sudo systemctl restart redis
```

Now, connect with the client again and confirm that your test value is still available:

```
$ redis-cli
```

```
127.0.0.1:6379> get test
```

The value of your key should still be accessible:

```
Output
"It's working!"
```

Back out into the shell again when you are finished:

```
127.0.0.1:6379> exit
```

Enable Redis to Start at Boot

If all of your tests worked, and you would like to start Redis automatically when your server boots, you can enable the systemd service.

To do so, type:

```
$ sudo systemctl enable redis
```

```
Output
Created symlink from /etc/systemd/system/multi-user.target.wants/redis.service to /etc/systemc
```

Conclusion

You should now have a Redis instance installed and configured on your Ubuntu 16.04 server. To learn more about how to secure your Redis installation, take a look at our How To Secure Your Redis Installation on Ubuntu 14.04 (from step 3 onward). Although it was written with Ubuntu 14.04 in mind, it should mostly work for 16.04 as well.

By: Justin Ellingwood

Related Tutorials

How To Install and Secure Redis on Debian 9

How To Install Redis from Source on Ubuntu 18.04

How to Secure Your Redis Installation on Ubuntu 18.04

How to Install MongoDB on Ubuntu 18.04

How To Install and Secure Redis on Ubuntu 18.04

60 Comments

Leave a comment...

Log In to Comment

barryso *May 29, 2016*

1 That was helpful. Thanks!

---

ymeiner *June 10, 2016*

0 Super helpful!!! thank you!

---

martijnr *June 12, 2016*

5 Thanks for writing this up! I had to add `Type=forking` to the systemd `[Service]` and change the ownership of `/var/log/redis` (but that may have been my configuration). Otherwise it would shut the service down immediately after starting it.

---

fernandoayl *July 3, 2018*

0 Yes, I just followed this guide exactly and this fixed it.
If you can't start/restart the server try running "redis-server /etc/redis/6379.conf" at the terminal, it should tell you if something is wrong with the config

---

epapa *June 21, 2016*

0 Thanks a lot, this really helped.

---

rwingard *July 12, 2016*

0 Excellent article. Worked perfectly. Thanks!

CNSKnight *July 25, 2016*

1 FYI: If you are getting start failures and no messages to the effect at all and no log yet generated as in my case, It's probably something in the startup chain - eg the redis.config in my case. This will allow you to test the startup directly (yet w/no systemctl hooks:

```
$ sudo /usr/local/bin/redis-server /etc/redis/redis.conf
```

---

anderles *August 21, 2016*

0 It's maybe best instruction for redis setup. Thanks.

---

dvp0008 *September 2, 2016*

0 I cannot connect it from my laptop, using "redis-cli -h IP.ADDRS.OF.DO.SERVER -p 6379" Do I have to enable something ?

---

kololobo *September 7, 2016*

2 /tmp/redis-stable$ sudo make install
Hint: It's a good idea to run 'make test' ;)

```
INSTALL install
INSTALL install
INSTALL install
INSTALL install
INSTALL install
```

---

mfhadil05 *January 24, 2017*

1 I also got this problem. so leave the tutorial and use $ sudo apt-get install redis-server instead?

---

rizki16d8f2c4c44ffaa52fe5d *July 10, 2017*

1 @mfhadil05 @kololobo I just got the same problem. I don't know why, I try to check the redis by simply typing "service redis status", I got redis service status running, then I just try to go ahead the tutorial without "sudo make install", it works just fine.

---

martinmiels *October 2, 2016*

4 Of course, you could always just

$ sudo apt-get install redis-server

---

chrisagiddings *October 23, 2016*

0 Thank you for this. I was having the same issue @kololobo was having.

This is so much simpler than what this walkthrough suggests.

---

fernandosouza84 *October 17, 2016*

0 Perfect. Thanks. :-D

AlexPolymath *October 27, 2016*

5 I followed the guide, but I got this.

```
● redis.service - Redis In-Memory Data Store
   Loaded: loaded (/etc/systemd/system/redis.service; disabled; vendor preset: enable
   Active: inactive (dead)

Oct 27 12:54:39 cs51351 systemd[1]: Stopped Redis In-Memory Data Store.
Oct 27 12:54:39 cs51351 systemd[1]: Started Redis In-Memory Data Store.
Oct 27 12:54:39 cs51351 systemd[1]: redis.service: Main process exited, code=exited,
Oct 27 12:54:39 cs51351 systemd[1]: redis.service: Control process exited, code=exite
Oct 27 12:54:39 cs51351 systemd[1]: redis.service: Unit entered failed state.
Oct 27 12:54:39 cs51351 systemd[1]: redis.service: Failed with result 'exit-code'.
Oct 27 12:54:39 cs51351 systemd[1]: redis.service: Service hold-off time over, schedu
Oct 27 12:54:39 cs51351 systemd[1]: Stopped Redis In-Memory Data Store.
Oct 27 12:54:39 cs51351 systemd[1]: redis.service: Start request repeated too quickly
Oct 27 12:54:39 cs51351 systemd[1]: Failed to start Redis In-Memory Data Store.
```

---

dkearney *November 8, 2016*

0 same problem, i missed **sudo make install**

---

vijayedkal *November 11, 2016*

1 Hi, I am also having same problem as Alexander of getting issue of "Redis In-Memory Data Store, Failed to start Redis In-Memory Data Store".

Can any one help us with any suggestions?

kuznecov513 *January 13, 2017*

0 the same

samelh *March 22, 2017*

1 Thanks to dkearney for the hint, we all missed `sudo make install` , and rolling back the install fixed it for me. http://grainier.net/how-to-uninstall-redis-server-from-ubuntu/

mihaiirimies *October 31, 2016*

0 Thx for this tutorial. Really helped me altot. Kudos.

fabiantjoeaon *November 25, 2016*

0 Working like a charm, thanks man :)!

dytpt *December 2, 2016*

3 thanks for you article, but when i run **sudo systemctl enable redis**, i got a error **Failed to execute operation: Invalid argument**.
Please help to fix it.

havlicekudanax *February 2, 2017*

0 [deleted]

havlicekudanax *February 2, 2017*

0 I had the same issue and the reason was not completely copied /etc/systemd/system/redis.service

mikee *January 17, 2018*

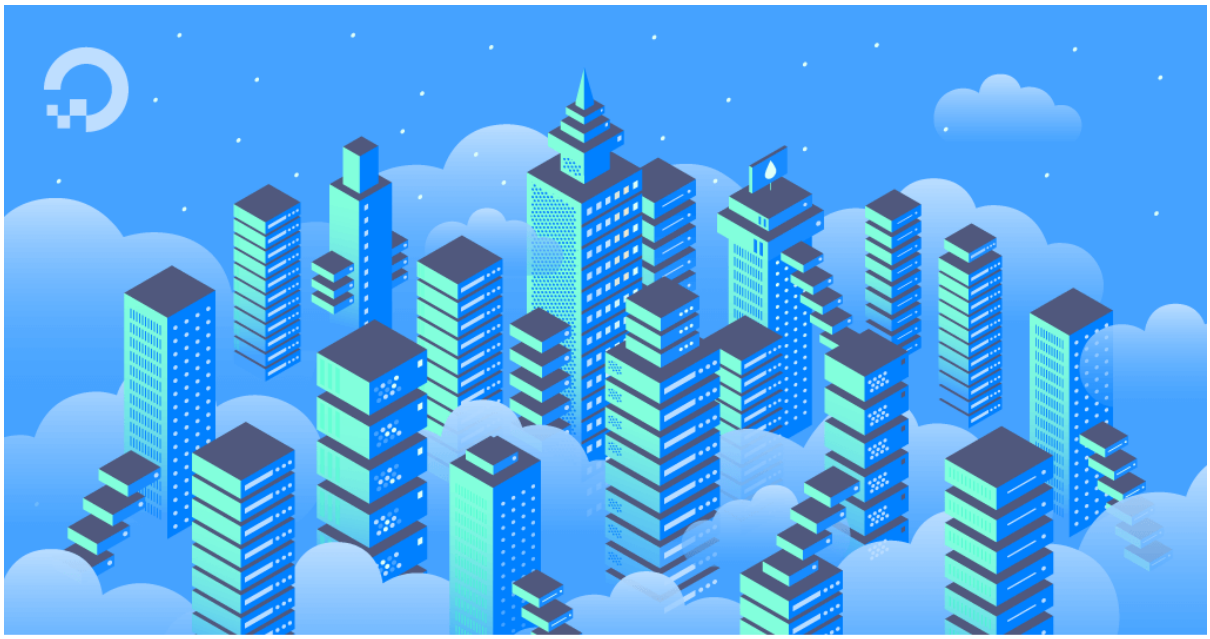
0 I just got the same problem.

alakdeb *December 12, 2016*

0 When I'm using the following command `sudo systemctl start redis` to start the reds server, getting this error 'sudo: systemctl: command not found'. Do I have to enable something?

jellingwood MOD *December 12, 2016*

0 @alakdeb Hey there. If you're getting a message telling you that the `systemctl` command is not available, you are probably trying to follow this guide using a different version of Ubuntu. Ubuntu 16.04 ships with the systemd init system. Most older versions of Ubuntu use Upstart, which is an alternative that won't work with these steps. If you are using an older version of Ubuntu, you can try using this guide.



How To Install and Use Redis

by Etel Sverdlov

This tutorial covers the Redis Installation, some simple Redis operations, and the 5 Redis data types. Redis, developed in 2009, is a flexible, open-source, key value data store that allows users to store vast amounts of data without the limits of a relational database.

therealssj *January 16, 2017*

1 Instead of configuring redis manually you can just go to `./redis-stable/utils` and there use the `install_server.sh` shell script to configure it. Much more simple!

devoh *February 19, 2017*

0 Thanks DO for this tutorial. As several people have already encountered the same problem, thus it's not an isolated case, if you get this error :

```
*** FATAL CONFIG FILE ERROR ***
Reading the configuration file, at line XXX
```

Here a solution, after this line of code :

```
sudo chmod 770 /var/lib/redis
```

I had to add those 2 to change the privilege of the log file.

```
sudo touch /var/log/redis.log
sudo chown redis:redis /var/log/redis.log
```

I am not sure it's the best solution but it worked for me. I think the tutorial should be updated with the appropriate solution.

---

pratyum *March 1, 2017*

0 Thanks a lot ! Really great documentation

---

smartdamini *March 3, 2017*

0 Thank you so much. very very useful document.

Load More Comments