How to Install MongoDB on CentOS 8

July 15, 2020

CENTOS 8 MONGODB

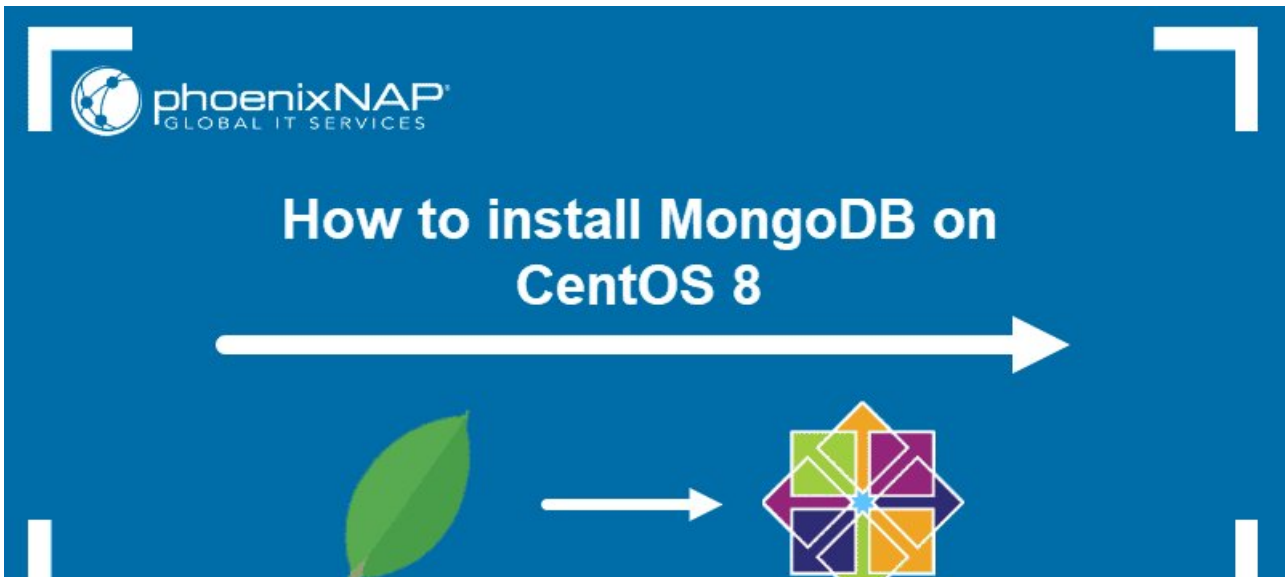
Contents

Introduction

MongoDB is a document-based NoSQL database application. Unlike MySQL, it allows data to be stored differently in different documents.

It allows for different fields in different documents, and the data structure is not permanently fixed.

In this tutorial, learn how to install MongoDB on CentOS 8.



Prerequisites

- A system running 64-bit CentOS 8 Linux
- User account with sudo or root privileges
- Access to a terminal window/command line

Installing MongoDB on CentOS 8

Step 1: Add the MongoDB Software Repository

By default, MongoDB is not available in the official CentOS repositories. To add the MongoDB repositories, open a terminal window, and create a MongoDB repository configuration file:

```
sudo nano /etc/yum.repos.d/mongodb-org-4.2.repo
```

In the newly created repo configuration file, enter the following:

```
[mongodb-org-4.2]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/
4.2/x86_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-4.2.asc
```

Save the file (**Ctrl+o**) and exit (**Ctrl+x**).

Note: At the time this article was written, MongoDB 4.2 was the latest version. Please check the MongoDB developer page for the latest version.

Step 2: Install MongoDB Software

Install MongoDB on CentOS 8 with the following command:

```
sudo yum install -y mongodb-org
```

```
Installed:
  mongodb-org-4.2.8-1.el8.x86_64
  python2-pip-9.0.3-16.module_el8.2.0+381+9a5b3c3b.noarch
  python2-setuptools-39.0.1-11.module_el8.2.0+381+9a5b3c3b.noarch
  python2-2.7.17-1.module_el8.2.0+381+9a5b3c3b.x86_64
  python2-libs-2.7.17-1.module_el8.2.0+381+9a5b3c3b.x86_64
  python2-pip-wheel-9.0.3-16.module_el8.2.0+381+9a5b3c3b.noarch
  python2-setuptools-wheel-39.0.1-11.module_el8.2.0+381+9a5b3c3b.noarch
  mongodb-org-mongos-4.2.8-1.el8.x86_64
  mongodb-org-server-4.2.8-1.el8.x86_64
  mongodb-org-shell-4.2.8-1.el8.x86_64
  mongodb-org-tools-4.2.8-1.el8.x86_64

Complete!
```

Step 3: Start the MongoDB Service

Start the MongoDB service by entering the following command:

```
sudo systemctl start mongod
```

If you receive an error that the unit is not found, run the following command, then try the previous command again:

```
sudo systemctl daemon-reload
```

If you're using MongoDB as a permanent feature, you can set it to run at boot with the following command:

```
sudo systemctl enable mongod
```

To check whether the MongoDB service is running, use the following command:

```
sudo systemctl status mongod
```

```
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor pres>
   Active: active (running) since Wed 2020-07-01 07:06:33 EDT; 18min ago
     Docs: https://docs.mongodb.org/manual
 Main PID: 7478 (mongod)
   Memory: 77.7M
   CGroup: /system.slice/mongod.service
           └─7478 /usr/bin/mongod -f /etc/mongod.conf

Jul 01 07:06:31 localhost.localdomain systemd[1]: Starting MongoDB Database Ser>
Jul 01 07:06:31 localhost.localdomain mongod[7476]: about to fork child process>
Jul 01 07:06:31 localhost.localdomain mongod[7476]: forked process: 7478
Jul 01 07:06:33 localhost.localdomain mongod[7476]: child process started succe>
Jul 01 07:06:33 localhost.localdomain systemd[1]: Started MongoDB Database Serv>
lines 1-14/14 (END)
```

Set Up and Configure MongoDB

Create MongoDB Admin User

Start by opening the Mongo shell for use. Enter the following command:

```
mongo
```

The prompt should change to a simple angle bracket.

```
>
```

Next, switch to the admin user account:

```
use admin
```

```
> use admin
switched to db admin
```

Next, create an administrator user account for the Mongo database:

```
db.createUser(
 {
 user: "mdbadmin",
```

```
  pwd: "password",
  roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
  }
  )
```

The system should respond with the following:

```
Successfully added user: {
        "user" : "mdbadmin",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                }
        ]
}
>
```

Note: Replace `mdbadmin` with an actual administrator username you want to use. Also, replace `password` with a unique strong, and secure password.

Next, display the list of users:

```
show users
```

The system should display details about the username just created:

```
{
        "_id" : "admin.mdbadmin",
        "userId" : UUID("ab3659e9-df0b-4484-a602-1e882acde869"),
        "user" : "mdbadmin",
        "db" : "admin",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                }
        ],
        "mechanisms" : [
                "SCRAM-SHA-1",
                "SCRAM-SHA-256"
        ]
}
```

Exit the Mongo shell by entering the following:

```
quit()
```

Remember, the alphanumeric **userId** will be different from this example.

Configure MongoDB Authentication

By default, any user can perform any function in MongoDB. This will require users to have proper credentials to perform actions.

Step 1: Turn on Authentication

Start by editing the following file:

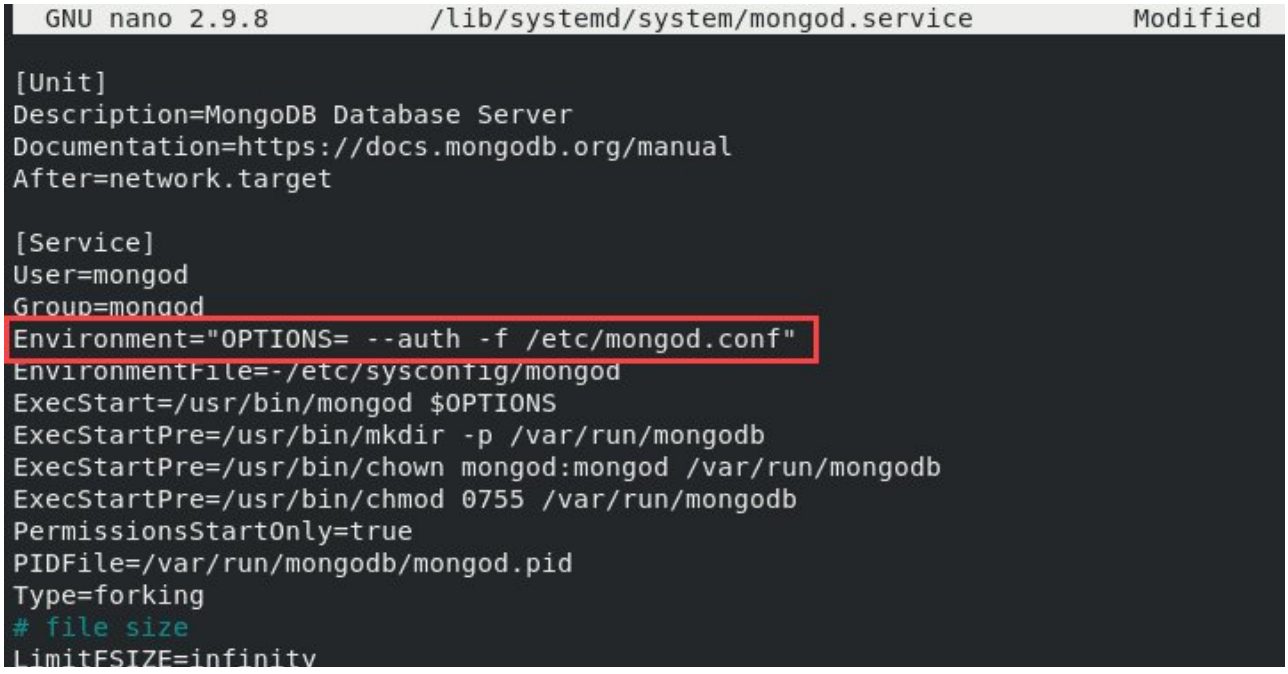
```
sudo nano /lib/systemd/system/mongod.service
```

Find the following line:

```
Environment="OPTIONS=--f /etc/mongod.conf"
```

Add the `--auth` option as follows:

```
Environment="OPTIONS= --auth -f /etc/mongod.conf"
```



Save the file (**Ctrl+o**) and exit (**Ctrl+x**).

Step 2: Reload the Services to Apply Changes

Reload the mongod.service:

```
sudo systemctl --system daemon-reload
```

```
sudo systemctl restart mongod
```

Step 3: Test Mongo User Authentication

Switch to the Mongo shell and use the admin user to list all users:

```
mongo
```

```
use admin
```

```
show users
```

An error message should display:

```
2020-07-01T08:30:48.302-0400 E  QUERY    [js] uncaught exception: Error: command
 usersInfo requires authentication :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.getUsers@src/mongo/shell/db.js:1638:15
shellHelper.show@src/mongo/shell/utils.js:883:9
shellHelper@src/mongo/shell/utils.js:790:15
@(shellhelp2):1:1
```

Next, use the following command to authenticate with the credentials created in **Part 2**:

```
db.auth('mdbadmin', 'password')
```

The system should respond with the number 1:

```
1
```

```
> db.auth('mdbadmin', 'password')
1
```

Now, try running the `show users` command again:

```
show users
```

Replace `mdbadmin` and `password` with the actual username and password you created. The system should display the same user information as before in **Part 2**.

Conclusion

You should now have a working installation of MongoDB on your CentOS 8 system. Also, you should have a secure administrator account to prevent unauthorized access.

Next, learn how to create a database in MongoDB.

Sofija Simic

Sofija Simic is an aspiring Technical Writer at phoenixNAP. Alongside her educational background in teaching and writing, she has had a lifelong passion for information technology. She is committed to unscrambling confusing IT concepts and streamlining intricate software installations.

Next you should read

Databases, SysAdmin

How to Install MongoDB on Ubuntu 18.04

November 26, 2019

MongoDB is a database program that provides high performance, high availability, and automatic scaling to...

READ MORE

Databases, Virtualization

How to Deploy and Manage MongoDB with Docker

February 25, 2020

This tutorial shows you how to use Docker and an official MongoDB container image to deploy your databases.

READ MORE

Databases, DevOps and Development

Cassandra vs MongoDB - What are the Differences?

April 27, 2020

Learn about the difference between Cassandra and MongoDB. These NoSQL databases have some similarities, but...

READ MORE

Bare Metal Servers, Databases, Web Servers

What Is NoSQL Database? - NoSQL Explained

June 17, 2020

The article provides a detailed explanation of what a NoSQL databases is and how it differs from relational...

READ MORE