## Step 1 – Adding the MongoDB Repository

The mongodb-org package does not exist within the default repositories for CentOS. However, MongoDB maintains a dedicated repository. Let's add it to our server.

Use any editor to open the following file. Here, the example is shown with the vi editor. The example describes the process to create a .repo file for yum, the package management utility for CentOS:

**sudo vi /etc/yum.repos.d/mongodb-org.repo**

**(As you already know, you can skip using sudo, if you are a root user).**

After that, visit the **<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/#configure-the-package-management-system-yum>** link for knowing the current stable version of mongodb. So, add the repository information for the latest stable release to the file opened (mongodb-org.repo). The time when I installed it, the current stable version was mongodb 3.4

**[mongodb-org-3.4]**

**name=MongoDB Repository**

**baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.4/x86\_64/**

**gpgcheck=1**

**enabled=1**

**gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc**

Save and close the file.

Before moving on, whether the MongoDB repository exists within the yum utility or not must be verified. The repolist command displays a list of enabled repositories:

**yum repolist**

Now, when you enter the following command, you will see the following line in the generated list if mongoDB repository exists within the yum utlity.

**mongodb-org-3.2/7/x86\_64 MongoDB Repository**

## Step 2 – Installing MongoDB

We can install the mongodb-org package from the third-party repository using the yum utility.

**sudo yum install mongodb-org**

There are two Is this ok [y/N]: prompts. The first one permits the installation of the MongoDB packages and the second one imports a GPG key. The publisher of MongoDB signs their software and yum uses a key to confirm the integrity of the downloaded packages. At each prompt, type Y and then press the ENTER key.

Next, start the MongoDB service with the systemctl utility:

**sudo systemctl start mongod**

Although we will not use them in this tutorial, you can also change the state of the MongoDB service with the reload and stop commands.

The reload command requests that the mongod process reads the configuration file, /etc/mongod.conf, and applies any changes without requiring a restart.

**sudo systemctl reload mongod**

The stop command halts all running mongod processes.

**sudo systemctl stop mongod**

The systemctl utility did not provide a result after executing the start command, but we can check that the service started by viewing the end of the mongod.log file with the tail command:

**sudo tail /var/log/mongodb/mongod.log**

You will see the following output:

**Output  
. . .[initandlisten] waiting for connections on port 27017**

An output of waiting for a connection confirms that MongoDB has started successfully and we can access the database server with the MongoDB Shell:

**mongo**

**Note:**

When you launched the MongoDB Shell you may have seen a warning like this:

\*\* WARNING: soft rlimits too low. rlimits set to 4096 processes, 64000 files. Number of processes should be at least 32000 : 0.5 times number of files.

MongoDB is a threaded application. It can launch additional processes to handle its workload. The warning states that for MongoDB to be most effective the number of processes that it is authorized to spin up should be half that of the number of files that it can have open at any given time. To resolve the warning, alter the processes soft rlimit value for mongod by editing the 20-nproc.conf file:

**sudo vi /etc/security/limits.d/20-nproc.conf**

Add the following line to the end of file **/etc/security/limits.d/20-nproc.conf**

**mongod soft nproc 32000**

For the new limit to be available to MongoDB, restart it using the systemctl utility:

**sudo systemctl restart mongod**

Afterward, when you connect to the MongoDB Shell, the warning should cease to exist.