

1. Importando a chave pública utilizada pelo gerenciamento de pacotes

Abra o terminal e utilize o comando abaixo para importar chave pública GPG do [MongoDB](#).

[Copiar](#)

```
wget -q0 - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key
```

Esse comando deve retornar um **OK**.

Porém, se você receber um erro indicando que **gnupg** não está instalado, faça como abaixo:

1. Instalar o **gnupg** e as bibliotecas necessárias através do comando:

[Copiar](#)

```
sudo apt-get install gnupg
```

2. Após a instalação, tente importar a chave outra vez:

[Copiar](#)

```
wget -q0 - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key
```

2. Crie o arquivo de lista (*list file*) para o MongoDB

Crie o arquivo `/etc/apt/sources.list.d/mongodb-org-4.2.list` para o Ubuntu 18.04 (Bionic):

[Copiar](#)

```
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-
```

3. Recarregue o banco de dados local de pacotes

[Copiar](#)

```
sudo apt-get update
```

4. Instale os pacotes do MongoDB

Você pode instalar a última versão estável do MongoDB ou uma versão específica.

Para instalar a última versão estável utilize o comando abaixo:

[Copiar](#)

```
sudo apt-get install -y mongodb-org
```

Pronto! Agora você já tem a última versão estável do **MongoDB** instalada e pronta para ser executada, mas antes vamos dar uma olhada nos pacotes que foram instalados.

Os Pacotes instalados

Você deve ter notado que durante a instalação alguns pacotes adicionais foram instalados:

1. **mongodb-org-server**: Esse pacote contém o que podemos chamar de "servidor" do MongoDB. Contém todos os recursos necessários para que uma instância do banco seja executada;
2. **mongodb-org-shell**: o *shell* é onde você se conecta com o MongoDB, através do terminal. É uma interface que suporta **JavaScript** e é super completa para administração de instâncias e clusters;
3. **mongodb-org-mongos**: Se pronuncia "Mongo S" e só se faz necessário em ambientes *Shard*. Não entraremos em detalhes sobre ele agora;
4. **mongodb-org-tools**: Esse pacote contém algumas ferramentas nativas do MongoDB. Por exemplo:
 - **mongodump**: ferramenta para extrair dados no formato **BSON** (falaremos deles mais adiante). Em alguns ambientes pode fazer parte da estratégia de backup
 - **mongorestore**: ferramenta para restaurar backups gerados pelo *mongodump*
 - **mongoimport**: ferramenta para importar arquivos **JSON**, **CSV** ou **TSV** para uma instância do MongoDB
 - **mongoexport**: exporta dados de uma instância do MongoDB para arquivos **JSON** ou **CSV**

Executando o MongoDB Community Edition

Considerações sobre o *ulimit*

Alguns sistemas operacionais baseados em *UNIX* limitam os recursos de sistema que uma sessão pode utilizar. Esses limites têm um grande impacto negativo para a operação do MongoDB e em ambientes de produção devem ser observados com muita atenção. Veja a seção [UNIX ulimit Settings](#) da documentação do MongoDB para maiores informações.

Diretórios de trabalho

Se você instalou o MongoDB via **apt** (gerenciador de pacotes), então algumas configurações são executadas e mantidas em diretórios do sistema operacional. Por

default, no Linux, os dados ficarão armazenados em `/var/lib/mongodb` e o log de funcionamento, em `/var/log/mongodb`. No MacOS, os dados e os logs ficam em `/usr/local/var/mongodb` e `/usr/local/var/log/mongodb`, respectivamente.

Por padrão, o MongoDB roda utilizando a conta de usuário `mongodb`, que também foi criada durante a instalação. Se você quiser rodar uma instância com outro usuário, deverá dar as permissões para ele nos diretórios de dados e log.

Arquivo de configuração

O pacote oficial inclui um [arquivo de configuração](#) (`/etc/mongod.conf`). Essas configurações (como especificação dos caminhos dos diretórios de dados e log) têm efeito após o *startup* da instância. Logo, se você fizer qualquer modificação nesse arquivo com a instância do MongoDB rodando, deverá reiniciá-la para que tenha efeito.

Mãos à obra, vamos executar!

1. Iniciando o MongoDB

No Linux:

[Copiar](#)

```
sudo service mongod start
```

No MacOS:

[Copiar](#)

```
brew services start mongodb-community
```

2. Verifique se o MongoDB foi iniciado com sucesso

No Linux:

[Copiar](#)

```
sudo service mongod status
```

No MacOS:

[Copiar](#)

```
brew services list | grep mongodb-community
```

Você também pode checar o arquivo de log que, por default, é localizado em `/var/log/mongodb/mongod.log`, no Linux, ou em `/usr/local/var/log/mongodb`, no Mac. Você pode verificar se a instância está rodando e pronta para conexões através da linha abaixo:

```
[initandlisten] waiting for connections on port 27017
```

Parando a instância

No Linux:

[Copiar](#)

```
sudo service mongod stop
```

No MacOS:

[Copiar](#)

```
brew services stop mongodb-community
```

Reiniciando a instância

No Linux:

[Copiar](#)

```
sudo service mongod restart
```

No MacOS:

[Copiar](#)

```
brew services restart mongodb-community
```

Informação importante

Por *default*, o MongoDB só permite conexões locais, ou seja, apenas de *clients* que estejam rodando na mesma máquina onde a instância estiver sendo executada. Para alterar essa configuração e permitir conexões remotas, você tem duas opções:

- no arquivo de configuração do MongoDB, com a opção [bindIp](#), ou
- via linha de comando, passando o argumento `--bind_ip`

Para mais informações sobre como configurar, veja [IP Binding](#).

Databases, Coleções e Documentos

O **MongoDB** armazena [documentos BSON](#) dentro de coleções e coleções dentro de *databases*.

