

## mongodb

Posted February 22nd, 2024

Mongodb is a popular nosql server  
MongoDB

<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-amazon/>

To install:

a) set up yum repo

/etc/yum.repos.d/mongodb-org-7.0.repo file should contain

```
#####
```

```
[mongodb-org-7.0]
```

```
name=MongoDB Repository
```

```
baseurl=https://repo.mongodb.org/yum/amazon/2023/mongodb-org/7.0/x86_64/
```

```
gpgcheck=1
```

```
enabled=1
```

```
gpgkey=https://pgp.mongodb.com/server-7.0.asc
```

```
#####
```

b) mongodb runs a user mongod, systemctl start mongod

/var/lib/mongo is repository

/var/log/mongodb is logging directory

c) mongosh = mongo shell

```
test> use mydb // change to database
```

```
mydb> show collections // show collections
```

```
db.users.insertOne({'username'=>'triemer'});
```

```
db.users.find();
```

d) So install on amazon 2023 doesn't work for mongosh

```
> dnf remove mongodb-mongosh
```

```
> dnf install -qy mongodb-mongosh-shared-openssl3
```

e) For PHP, install mongodb extension

```
-- Install the mongodb extension to php
```

```
>pecl install mongodb
```

```
(I chose all the defaults)
```

```
> cd /usr/local/lib/php/extensions
```

```
mongodb built in subdir... copy up one level
```

```
>add extension=mongodb.so to php.ini (/usr/local/lib/php.ini)
```

```
-- now install php library
```

```
in your php project where composer.json is
```

```
composer require mongodb/mongodb
```

f) basic code to initiate mongodb from php

From: <https://www.mongodb.com/docs/php-library/current/tutorial/crud/>

```
<?php
```

```
require_once "lib/vendor/autoload.php";
```

```
$serverApi = new ServerApi(ServerApi::V1);
```

```
$client = new MongoDB\Driver\Client('mongodb://127.0.0.1',[],['serverApi'=>$serverApi]);
```

CRUD operations:

define the collection:

```
$collection = $client->test->users ( creates a users table in test database)
```

```
// CREATE
```

```
$result = $client->insertOne(['username'=>'admin',email=>admin@fubar.com']);
```

```
Get Count: echo $result->getInsertedCount();
```

```
Get Inserted Id: echo $result->getInsertedId();
```

```
$result = $client->insertMany(
```

```
    ['username'=>'admin','email'=>'admin@fubar.com'],
```

```
    ['username'=>'tom','email'=>'tom@fubar.com']
```

```
);
```

```
Get Count: echo $result->getInsertedCount();
```

```
Get Inserted Ids: echo $result->getInsertedIds();
```

```
// SERACH
```

```

$cursor = $collection->find(['username'=>'admin',
    'last_name'=> MongoDB\BSON\Regex('^garden', 'i')
], $options);
// Mix and match options
$options = ['projection'=>
    ['username'=>1,
    'first_name'=>1,
    'last_name'=>1],
    'limit'=>4,
    'sort'=>['pop' => -1],
    'sort'=>
];

```

## Update

```

-----
$result = $collection->updateOne(
    ['name'=>'Tom']
    ['$set'=>['address'=>'123 test way']]
);
$count = $result->getMatchedCount();
$modified = $result->getModifiedCount();

$result = $collection->updateMany(
    ['state'=>'ny'],
    ['$set'=>['country'=>'us']]);
** Updated only updates the fields mentioned **

```

## Replace

```

-----
$result = $collection->replaceOne(
    ['name'=>'Tom'], // filter
    ['username'=>'thomas', // new document
    'last_name'=>'myname'];
$count = $result->getMatchedCount();
$modified = $result->getModifiedCount();

```

/\* This replaces all fields \*/

## Upsert

```

-----
$updateResult = $collection->updateOne(
    ['name' => 'Bob'],
    ['$set' => ['state' => 'ny']],
    ['upsert' => true]
);
** updates the table if exists, creates if not */

```

## Delete

```

-----
$collection->deleteOne(['name'=>'bob']);
$collection->deleteMany(['name'=>'tom']);

```

## Indexes

```

$collection->createIndex(['name'=>1]);
foreach ($collection->listIndexes() as $indexInfo) {
    var_dump($indexInfo);
}
$collection->dropIndex();
$collection->dropIndexes(); // drops all indexes on collection

```

-----  
installation on rhel 8.9

-----  
<https://linuxconfig.org/how-to-install-mongodb-on-redhat-8>

Then follow commands above to install pecl and library

-----  
Enable authentication

-----  
1. create an admin user  
use admin

```
createUser( { user:"myUserAdmin", pwd:"mypath",
  roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
});
```

2. configure /etc/mongod.conf

```
Security:
  authorization:enabled
```

3. now create user in the database

```
use admin
db.auth("myUserAdmin","mypath")
```

```
use mydb
```

```
createUser( { user:"fancyuser", pwd:"mypass",
  roles: [ { role: "readWrite", db:"mydb"}]
})
```

4. in php configure the url for the collection connection

```
$collection = new MongoClient('mongodb://fancyuser:mypass@127.0.0.1/mydb');
e.g. mongodb://user:password@127.0.0.1/database
```

-----  
Creating a database  
-----

Bit weird here...

1. authorize with myUserAdmin

```
> use admin
> db.auth("myUserAdmin","mypath")
> use newdb
> createUser( { user:"fancyuser", pwd:"mypass",
  roles: [ { role: "readWrite", db:"newdb"}]
})
```

2. the database will not show up in user databases until something is written to it.

3. BUT the authorization works.