

Mongodb+python

Login to mongodb



Create a database

Choose your cloud provider, region, and specs.

Build a Database

Once your database is up and running, live migrate an existing MongoDB database into Atlas with our [Live Migration Service](#).

Click on built a database

MongoDB.
MONGODB ATLAS

Deploy a cloud database

Experience the best of MongoDB on AWS, Azure, and Google Cloud. Choose a deployment option to get started.

NEW	ADVANCED	FREE
Serverless For application development and testing, or workloads with variable traffic. Minimal configuration required. <ul style="list-style-type: none">✓ Pay only for the operations you run✓ Resources scale seamlessly to meet your workload✓ Always-on security and backups Create Starting at \$0.10/1M reads	Dedicated For production applications with sophisticated workload requirements. Advanced configuration controls. <ul style="list-style-type: none">✓ Network isolation and fine-grained access controls✓ On-demand performance advice✓ Multi-region and multi-cloud options available Create Starting at \$0.08/hr* <small>*estimated cost \$56.94/month</small>	Shared For learning and exploring MongoDB in a cloud environment. Basic configuration options. <ul style="list-style-type: none">✓ No credit card required to start✓ Explore with sample datasets✓ Upgrade to dedicated clusters for full functionality Create Starting at FREE

Click on create in free shared

Scroll down and find create cluster and click on it

Now give a simple username and password.... which does not have a special character

Then click on create user

Now add ip address and description as shown below

Add entries to your IP Access List

Only an IP address you add to your Access List will be able to connect to your project's clusters.

IP Address	Description	
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="whitelist"/>	<input type="button" value="Add My Current IP Address"/>
<input type="button" value="Add Entry"/>		

Then click on add entry and then click on Finish and Close

Enter IP Address Enter description Add My Current IP Address

Add Entry

Access List

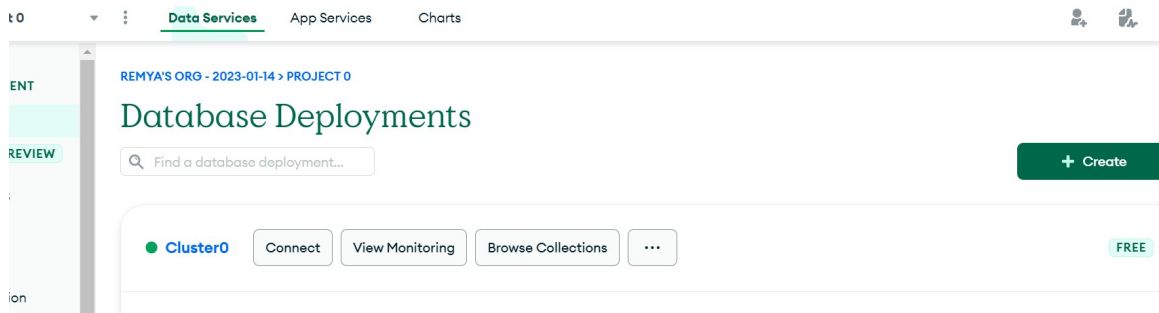
0.0.0.0

Congratulations on setting up access rules!

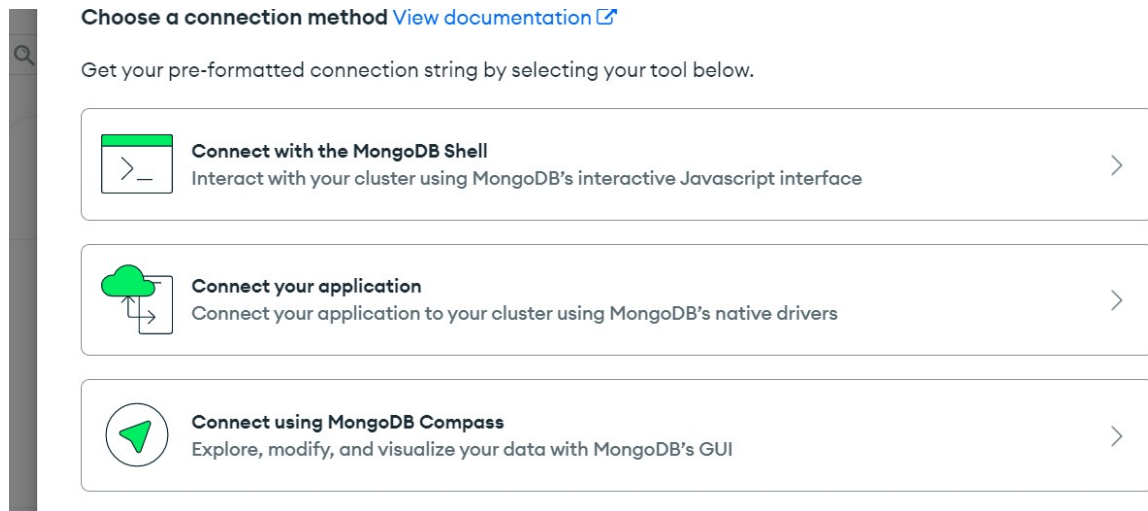
You will now be able to connect to your deployments. You can continue to add and update access rules in [Database Access](#) and [Network Access](#).

☒ Hide Quickstart guide in the navigation. You can visit [Project Settings](#) to access it in the future.

Now click on Go to Databases



Now click on connect



Click on connect to your application

Connect to Cluster0

✓ Setup connection security

✓ Choose a connection method

Connect

1 Select your driver and version

DRIVER

Python

VERSION

3.6 or later

2 Add your connection string into your application code

☒ Include full driver code example

```
client = pymongo.MongoClient("mongodb+srv://remya:
<password>@cluster0.gzq32uo.mongodb.net/?retryWrites=true&w=majority")
db = client.test
```

Replace **<password>** with the password for the **remya** user. Ensure any option params are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

Now choose python in deliver and 3.6 or later in version then select the url as shown above

Then click on close

Now open jupyter notebook and write following code

```
In [1]: ! pip install pymongo
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting pymongo
  Downloading pymongo-4.3.3-cp39-cp39-win_amd64.whl (382 kB)
    ----- 382.5/382.5 kB 3.4 MB/s eta 0:00:00
Collecting dnspython<3.0.0,>=1.16.0
  Downloading dnspython-2.2.1-py3-none-any.whl (269 kB)
    ----- 269.1/269.1 kB 16.2 MB/s eta 0:00:00
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.2.1 pymongo-4.3.3
Note: you may need to restart the kernel to use updated packages.
```

```
In [1]: ! import pymongo
```

```
client = pymongo.MongoClient("mongodb+srv://remya:remya@cluster0.gzq32uo.mongodb.net/?retryWrites=true&w=majority")
db = client.test
print(db)

Database(MongoClient(host=['ac-dhcudeq-shard-00-01.gzq32uo.mongodb.net:27017', 'ac-dhcudeq-shard-00-02.gzq32uo.mongodb.net:27017', 'ac-dhcudeq-shard-00-00.gzq32uo.mongodb.net:27017'], document_class=dict, tz_aware=False, connect=True, retrywrites=True, w='majority', authsource='admin', replicaset='atlas-39unyw-shard-0', tls=True), 'test')
```

in above code : client = pymongo..... line is actually the url copied edited with username and password

Now write the following code and execute them:

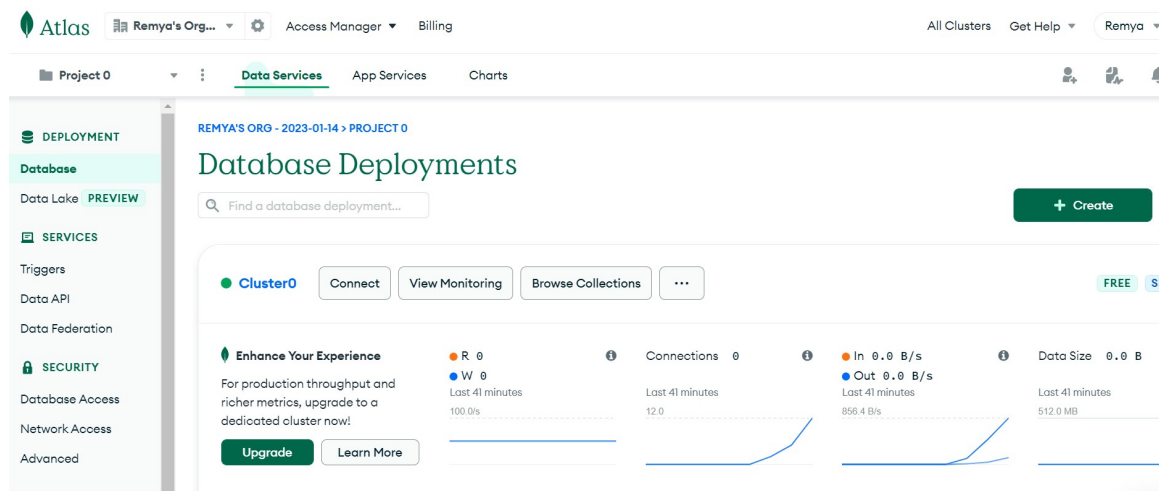
```
In [2]: db = client['ineuron']
col = db['course']
coll = db['students']

In [3]: data = {
    'course_name': ['full stack data science', 'full stack data analytics', 'big data', 'devops', 'blockchain'],
    'course_instructor': ['sudhanshu', 'anand', 'shashank', 'hitesh', 'navin'],
    'start_date': '12-12-2023',
    'duration': 'six months',
    'mode': 'live online',
    'certification': 'yes',
    'prerequisite': 'Dedication',
    'resume': 'yes',
    'mockinterview': 'yes'
}

In [4]: coll.insert_one(data)

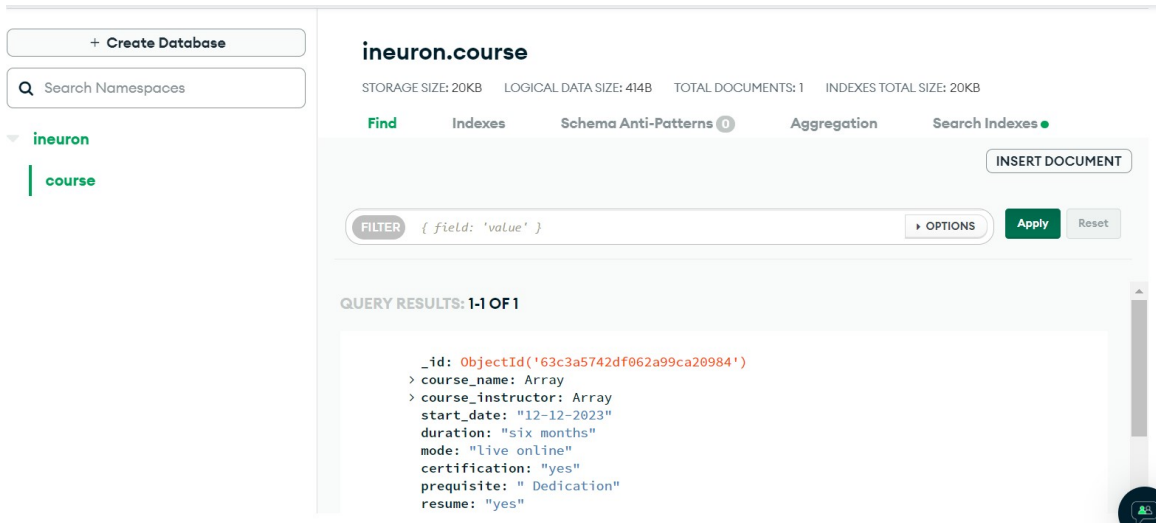
Out[4]: <pymongo.results.InsertOneResult at 0x226fa351820>
```

Now check mongodb...



Click on Browse collections

And find the row inserted recently in collection



We can insert another row of completely different schema in mongodb as shown below

```
In [5]: ► data1 = {
        'name' : 'full stack web dev' ,
        'instructor' : ['hitesh' , 'anurag'],
        'price ' : '17700',
        'duration' : 'eight month'
      }

In [6]: ► col.insert_one(data1)

Out[6]: <pymongo.results.InsertOneResult at 0x226fa351e20>
```

Now on refreshing mongodb.... you can find the new row also:

FILTER

{ field: 'value' }

resume: "yes"

mockinterview: "yes"

_id: ObjectId('63c3a7ca2df062a99ca20985')

name: "full stack web dev"

> instructor: Array

price : "17700"

duration: "eight month"

Also multiple rows can be inserted at a time as shown below:

```
In [7]: ► data2 = [{
    'name' : 'full stack web dev' ,
    'instructor' : ['hitesh' , 'anurag'],
    'price ' : '17700',
    'duration' : 'eight month'
},
{
    'name' : 'full stack web dev' ,
    'instructor' : ['hitesh' , 'anurag'],
    'price ' : '17700',
    'duration' : 'eight month'
},
{
    'name' : 'full stack web dev' ,
    'instructor' : ['hitesh' , 'anurag'],
    'price ' : '17700',
    'duration' : 'eight month'
},
{
    'name' : 'full stack web dev' ,
    'instructor' : ['hitesh' , 'anurag'],
    'price ' : '17700',
    'duration' : 'eight month'
}]
```

```

},
{
  'name' : 'full stack web dev' ,
  'instructor' : ['hitesh' , 'anurag'],
  'price ' : '17700',
  'duration' : 'eight month'
},
{
  'name' : 'full stack web dev' ,
  'instructor' : ['hitesh' , 'anurag'],
  'price ' : '17700',
  'duration' : 'eight month'
},
{
  'name' : 'full stack web dev' ,
  'instructor' : ['hitesh' , 'anurag'],
  'price ' : '17700',
  'duration' : 'eight month'
}
]

```

```
In [8]: ► col.insert_many(data2)
```

```
Out[8]: <pymongo.results.InsertManyResult at 0x226f81943a0>
```

Results updation of those rows in mongodb:


```
      _id: ObjectId('63c3a9572df062a99ca20986')
      name: "full stack web dev"
    > instructor: Array
      price : "17700"
      duration: "eight month"
```

```
      _id: ObjectId('63c3a9572df062a99ca20987')
```

Similarly more rows with different schema can be added:

```
In [9]: data3 = [{
      'name' : 'full stack web dev' ,
      'instructor' : ['hitesh' , 'anurag'],
      'price' : '17700',
      'duration' : 'eight month'
    } ,
    {
      'course_name' : ['full stack data science' , 'full stack data analytics' , 'big data' , 'devops' , 'blockchain'],
      'course_instructor' : ['sudhanshu' , 'anand' , 'shashank' , 'hitesh' , 'navin'],
      'start_date' : '12-12-2023',
      'duration' : 'six months',
      'mode' : 'live online',
      'certification' : 'yes',
      'prerequisite' : 'Dedication',
      'resume' : 'yes',
      'mockinterview' : 'yes'
    }
  ]
```

```
In [10]: col.insert_many(data3)
```

```
Out[10]: <pymongo.results.InsertManyResult at 0x226fa2be9a0>
```

Shows them in mongodb:

FILTER

{ field: 'value' }

▶ OPTIONS

Apply

Reset

```
_id: ObjectId('63c3aa712df062a99ca2098f')
> course_name: Array
> course_instructor: Array
start_date: "12-12-2023"
duration: "six months"
mode: "live online"
certification: "yes"
prerequisite: " Dedication"
resume: "yes"
mockinterview: "yes"
```

Rest coding is there in python_and_mongodb.ipynb

