

All about Mongoddb

Open MongoDB URL :


https://www.mongodb.com/cloud/atlas/lp/general/try?utm_source=compass&utm_medium=product

Configure Your Database using this simple steps

Database Access:

Database Access

Database Users Custom Roles



Create a Database User

Set up database users, permissions, and authentication credentials in order to connect to your clusters.

[Add New Database User](#)

[Learn more](#)

➔ Click on “Add New Database User “

Add New Database User

Create a database user to grant an application or user, access to databases and collections in your clusters in this Atlas project. Granular access control can be configured with default privileges or custom roles. You can grant access to an Atlas project or organization using the corresponding [Access Manager](#)

Authentication Method

Password

Certificate

AWS IAM
(MongoDB 4.4 and up)

MongoDB uses [SCRAM](#) as its default authentication method.

Password Authentication

[SHOW](#)

[Autogenerate Secure Password](#) [Copy](#)

Database User Privileges

Configure role based access control by assigning database users a mix of one built-in role, multiple custom roles, and multiple specific privileges. A user will gain access to all actions within the roles assigned to them, not just the actions those roles share in common. **You must choose at least one role or privilege.** [Learn more about roles.](#)

Built-in Role

Select one [built-in role](#) for this user.

Read and write to any database

1 SELECTED ^

Click on “Add User”

Database Access

Database UsersCustom Roles

+ ADD NEW DATABASE USER


User Name	Authentication Method	MongoDB Roles	Resources	Actions
chandrakant	SCRAM	readWriteAnyDatabase@admin	All Resources	EDIT DELETE

Network Access

- > Click on “Add IP Address”

Network Access

IP Access ListPeeringPrivate Endpoint



Add an IP address

Configure which IP addresses can access your cluster.

Add IP Address

[Learn more](#)

Database

Database Deployments

Find a database deployment...+ Create

Cluster0

ConnectView MonitoringBrowse Collections...

FREE SHARED

Enhance Your Experience

For production throughput and richer metrics, upgrade to a dedicated cluster now!

Upgrade

R 0

W 0

Last 6 hours

100.0%

Connections 0

Last 6 hours

100.0

In 0.0 B/s

Out 0.0 B/s

Last 6 hours

100.0 B/s

Data Size 0.0 B

Last 9 hours

912.0 MB

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SEARCH
5.0.14	AWS / Mumbai (ap-south-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Create Index

Click on -> Connet Your Application

Connect to Cluster0


✓ Setup connection security

Choose a connection method

Connect


Choose a connection method [View documentation](#)

Get your pre-formatted connection string by selecting your tool below.




Connect with the MongoDB Shell
Interact with your cluster using MongoDB's interactive Javascript interface

>




Connect your application
Connect your application to your cluster using MongoDB's native drivers

>



Connect using MongoDB Compass
Explore, modify, and visualize your data with MongoDB's GUI

>



Connect using VS Code
Connect to a MongoDB host in Visual Studio Code

>

Go Back

Close

Select Driver – Python

Version – 3.6 or Later

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://chandrakant:
<password>@cluster0.it5e7.mongodb.net/?retryWrites=true&w=majority")
db = client.test
```



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

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

Go Back

Close

In Jupyter Notebook

```
pip install pymongo
```

```
import pymongo
```

```
client = pymongo.MongoClient("mongodb+srv://chandrakant:chandrakant1@cluster0.it5e7.mongodb.net/?retryWrites=true&w=majority")  
db = client.test
```

```
pip install dnspython
```

Requirement already satisfied: dnspython in c:\programdata\anaconda3\lib\site-packages (2.2.1)Note: you may need to restart the kernel to use updated packages.

To Create a Database

```
database_name = "student" #data base name  
student_database = client[database_name]
```

To Create Collection

```
collection = "student_details"
```

```
student_detail_collection = student_database[collection]
```

To Create Document Inside Collection

```
student_data = {  
    "Name": "Chandrakant",  
    "College": "BoardInfinity",  
    "Address": "NaviMumbai"  
}
```

```
student_detail_collection.insert_one(student_data)
```

```
<pymongo.results.InsertOneResult at 0x2a3a5f6a3a0>
```

To Fetch Details from Mongoddb

```
student_cursor = student_detail_collection.find()
```

```
student_cursor.next()
```

```
{ '_id': ObjectId('63b0abb7236e0efb81dea3b0'),  
  'Name': 'Chandrakant',  
  'College': 'BoardInfinity',  
  'Address': 'NaviMumbai' }
```

```
for student_details in student_detail_collection.find():  
    print(student_details)
```

```
{ '_id': ObjectId('63b0abb7236e0efb81dea3b0'), 'Name': 'Chandrakant', 'College': 'BoardInfinity', 'Address': 'NaviMumbai' }
```

Insert Multiple Records in Database

```
student_data_list = [  
    {"First_name": "Chandrakant", "Last_name": "Thakur", "Designation": "Data Scientist"},  
    {"city": "Mumbai", "First_name": "Nilesh", "college": "BoardInfinity"},  
    {"First_name": "Binod", "address": "Jharhand", "college": "JH"}  
]
```

```
student_detail_collection.insert_many(student_data_list)
```

```
<pymongo.results.InsertManyResult at 0x2a3a8f3cb50>
```

```
for student_details in student_detail_collection.find():  
    print(student_details)
```

```
{ '_id': ObjectId('63b0abb7236e0efb81dea3b0'), 'Name': 'Chandrakant', 'College': 'BoardInfinity', 'Address': 'NaviMumbai' }  
{ '_id': ObjectId('63b0b41c76c1ce02db4534f6'), 'First_name': 'Chandrakant', 'Last_name': 'Thakur', 'Designation': 'Data Scientist' }  
{ '_id': ObjectId('63b0b41c76c1ce02db4534f7'), 'city': 'Mumbai', 'First_name': 'Nilesh', 'college': 'BoardInfinity' }  
{ '_id': ObjectId('63b0b41c76c1ce02db4534f8'), 'First_name': 'Binod', 'address': 'Jharhand', 'college': 'JH' }
```

How to Apply Filters

```
query = {"First_name": "Chandrakant"}
```

```
for student_details in student_detail_collection.find(query):  
    print(student_details)
```

```
{'_id': ObjectId('63b0b41c76c1ce02db4534f6'), 'First_name': 'Chandrakant', 'Last_name': 'Thakur', 'Designation': 'Data Scientist'}
```

Output:-

Final Output in MongoDB Atlas

The screenshot displays the MongoDB Atlas web interface for a database named 'student'. The collection 'student_details' is selected. At the top, statistics are shown: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 370B, TOTAL DOCUMENTS: 4, and INDEXES TOTAL SIZE: 36KB. Navigation tabs include Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. An 'INSERT DOCUMENT' button is in the top right. A filter bar contains the text '{ field: 'value' }' and buttons for 'OPTIONS', 'Apply', and 'Reset'. Below the filter, it says 'QUERY RESULTS: 1-4 OF 4'. Three document entries are visible:

- `_id: ObjectId('63b0abb7236e0efb81dea3b0')`
`Name: "Chandrakant"`
`College: "BoardInfinity"`
`Address: "NaviMumbai"`
- `_id: ObjectId('63b0b41c76c1ce02db4534f7')`
`city: "Mumbai"`
`First_name: "Nilesh"`
`college: "BoardInfinity"`
- `_id: ObjectId('63b0b41c76c1ce02db4534f6')`
`First_name: "Chandrakant"`
`Last_name: "Thakur"`
`Designation: "Data Scientist"`