

## EXPERIMENT NO.: 06

**Title:** Study and implementation of MongoDB cloud database

### **Solution:**

MongoDB is an open-source database management system (DBMS) that uses a document-oriented database model. MongoDB is written in C++. MongoDB supports various forms of data. MongoDB stores data in flat files using their own binary storage objects. This means that data storage is very compact and efficient, perfect for high data volumes. MongoDB stores data in JSON-like documents, which makes the database very flexible and scalable.

MongoDB is a document-oriented database model. Each MongoDB database contains collections and which in turn contains documents. Each document can be different and depends on the varying number of fields. The model of each document will be different in size and content from each other. The data model features allow you to store arrays and complex structured in a hierarchical relationship.

#### Characteristics of MongoDB

- **MongoDB is Schema-Less:** MongoDB is a schema-less database which flexible than traditional database tables. It is written in language C++. It has no schema to have many fields, content, and size different from another document in the same collection.
- **High Performance:** MongoDB is an open-source database with high performance. MongoDB is a high availability and scalability database. It supports faster query response because of features like indexing and replication.
- **MongoDB Indexing:** Indexing is very important for improving the performances of search queries. MongoDB uses indexing of dataset to enhance query performances and searches. MongoDB indexing enhances the performance for the faster search query. Document in a MongoDB can be used for indexing using primary and secondary indices.
- **File storage:** MongoDB can be used as a file system with load balancing and data replication features over multiple machines for storing files.
- **Replication:** The feature of replication is to distribute data to multiple nodes. It can have primary nodes and secondary nodes to replicate data. Replication of data is done using master-slave architecture. MongoDB provides a replication feature by distributing data across multiple machines.
- **Sharding:** This process distributes data across multiple physical partitions called shards, due to sharding MongoDB automatic process load balancing. We use sharding in cases where we need to work on very larger datasets.

#### Advantage of MongoDB

- **Flexible Document Schemas**

MongoDB's document model allows virtually any kind of data structure to be modeled and manipulated easily. MongoDB's BSON data format, inspired by JSON, allows you to have objects in one collection have different sets of fields (say, a middle name on a user only when applicable, or region-specific information that only applies to some records).

MongoDB supports creating explicit schemas and validating data so it doesn't get out of control, but this flexibility is an incredible asset when handling real-world data, and handling changes in requirements or environment.

- Code-native data access

Most databases force you to use heavy wrappers, like ORMs (Object Relational Mappers), to get data into Object form for use in programs. MongoDB's decision to store and represent data in a document format means that you can access it from any language, in data structures that are native to that language (e.g. dictionaries in Python, associative arrays in JavaScript, Maps in Java, etc.).

- Change-friendly design

If you're used to having to bring down your site or application in order to change the structure of your data, you're in luck: MongoDB is designed for change.

We spend a lot of time and effort designing efficient processes, and learning from our mistakes, but typically the database is slowing the whole thing down. There's no downtime required to change schemas, and you can start writing new data to MongoDB at any time, without disrupting its operations.

- Powerful querying and analytics

What good is a database if you can't find things inside it? MongoDB is designed to make data easy to access, and rarely to require joins or transactions, but when you need to do complex querying, it's more than up to the task.

The MongoDB Query Language (MQL) is a full-featured, powerful language that allows you to query deep into documents, and even perform complex analytics pipelines with just a few lines of JSON-like MQL.

- Easy horizontal scale-out

MongoDB is designed from the ground up to be a distributed database. Create clusters with real-time replication, and shard large or high-throughput collections across multiple clusters to sustain performance and scaler horizontally.

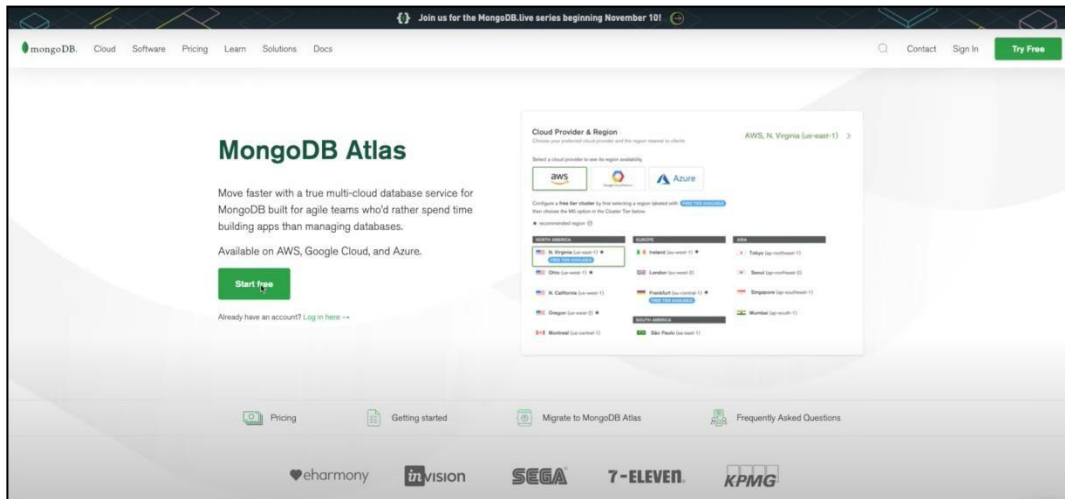
For implementation of MongoDB Database Cloud follows steps:

### **Step 1:**

Open the Follows URL in Web browser:

<https://www.mongodb.com/cloud/atlas>

web page will be open click into the "Start Free" or "Try Free" Button.



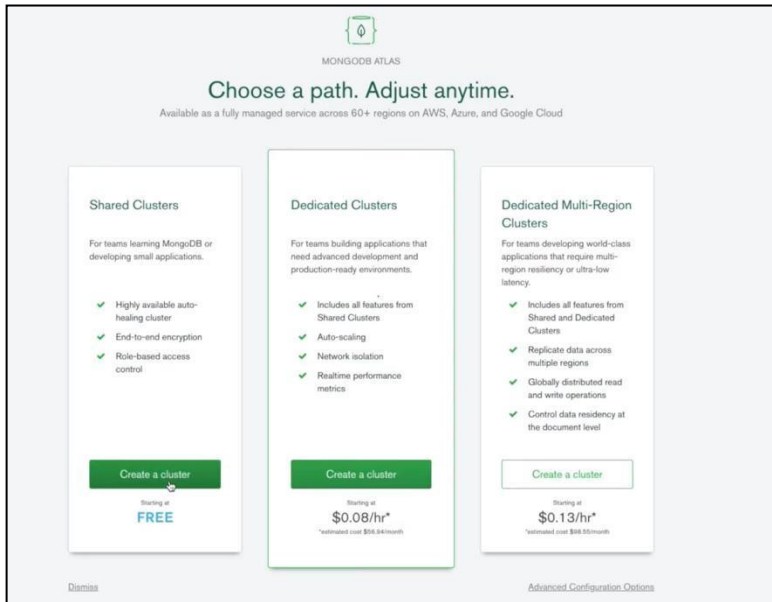
**Step 2:**  
**Create your MongoDB Atlas Account.**

**Step 3:**  
**Enter Your**

- 1. Organization name.**
- 2. Project name (for Create Project).**
- 3. Select the Programming language you referred.**

#### Step 4:

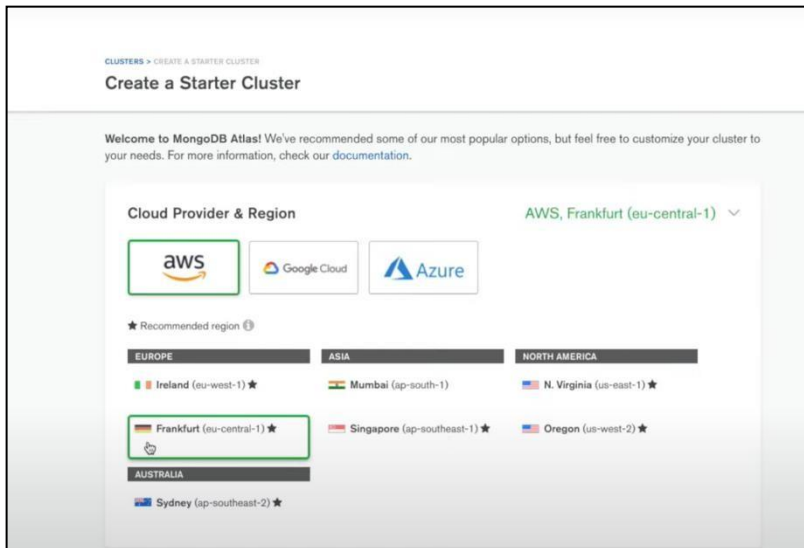
**Choose your Database Cluster Plan , for Practices Purpose select Free Plan**



#### Step 5:

**Create Starter Cluster**

**Select the Cloud Provider Company and Region**



## Step 6:

**Click on Cluster Tier and Select the M0 Sandbox (Free Version)  
And Change the Cluster Name Which name you need.**

The screenshot shows the 'Cluster Tier' selection screen in MongoDB Atlas. At the top, 'M0 Sandbox (Shared RAM, 512 MB Storage)' is selected, with a dropdown arrow and 'Encrypted' status. Below this, a note states: 'Base hourly rate is for a MongoDB replica set with 3 data bearing servers.' A section titled 'Shared Clusters for development environments and low-traffic applications' contains a table of cluster tiers.

Tier	RAM	Storage	vCPU	Base Price
<b>M0 Sandbox</b>	Shared	512 MB	Shared	Free forever
M2	Shared	2 GB	Shared	\$9 / MONTH
M5	Shared	5 GB	Shared	\$25 / MONTH

Below the table, a note says: 'M0 clusters are best for getting started, and are not suitable for production environments.' A summary bar lists: '500 max connections | Low network performance | 100 max databases | 500 max collections'.

The 'Additional Settings' section shows 'MongoDB 4.2, No Backup' selected. A toggle for 'Turn on Backup (M2 and up)' is set to 'NO'. A link 'See Backup Solutions for Paid Clusters (M2+)' is provided.

The 'Cluster Name' field is empty, with 'Cluster0' shown as a default or previous value.

At the bottom, a green 'Create Cluster' button is visible, along with a 'FREE' badge and a 'Back' link.

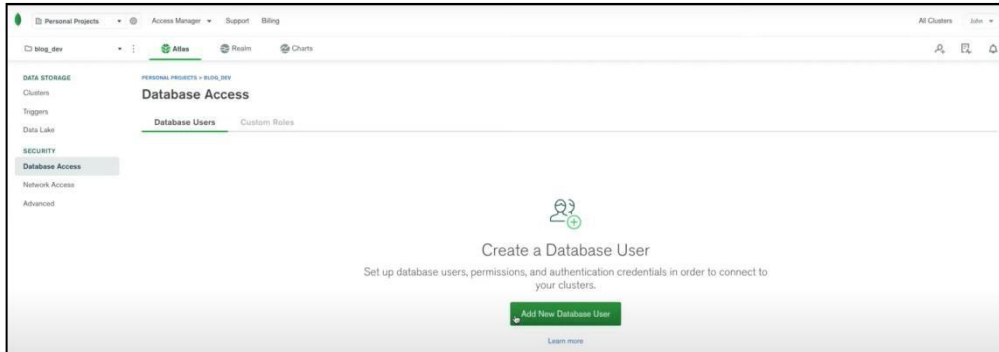
## Step 7:

The screenshot shows the 'Clusters' management page in MongoDB Atlas. The left sidebar has 'Database Access' selected under the 'Security' section. The main panel shows details for a cluster named 'Blog'. The 'Cluster Tier' is 'M0 Sandbox (General)'. The 'Region' is 'AWS - Frankfurt (eu-central-1)'. The 'Type' is 'Replica Set - 3 nodes'. The 'Unified Realm App' is 'None Linked'. The 'Atlas Search' is 'Create Index'. The 'Operations' tab is active, showing 'Read' and 'Write' operations. The 'Logical Size' is '0.0 B'. The 'Last 6 Hours' graph shows a flat line at 0.0 B. The 'Connections' tab shows '0' connections. The 'Last 6 Hours' graph shows a flat line at 0.0 B. An 'Upgrade' button is visible in the bottom right corner.

**Go to the Left Side Panel And click the 'Database Access' Option.**

## Step 8:

**For Create the New Database User Click Add New Database User Button.**



## Fill the Follows Forms

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 privileges or custom roles. You can grant access to an Atlas project or organization using the corresponding [Access Manager](#).

**Authentication Method**

☒ Password ☐ Certificate (M10 and up) ☐ AWS IAM (MongoDB 4.4 and up)

MongoDB uses SCRAM as its default authentication method.

**Password Authentication**

blog\_admin

SHOW

**Database User Privileges**

Select a built-in role or privileges for this user.

Read and write to any database

**Restrict Access to Specific Clusters/Data Lakes**

Enable to specify the resources this user can access. By default, all resources in this project are accessible. ☒

**Grant Access To**

Search user resources...

☐ Select all current Clusters

☒ Blog

☐ Select all current Data Lakes

No Data Lakes exist in this project

**Restrict Access to Specific Clusters/Data Lakes**

Enable to specify the resources this user can access. By default, all resources in this project are accessible. ☒ ON

**Grant Access To**

1 Cluster, 0 Data Lakes [Edit](#)

CLUSTERS

Blog

DATA LAKES

**Temporary User**

This user is temporary and will be deleted after your specified duration of 6 hours, 1 day, or 1 week. ☐ OFF

[Cancel](#) [Add User](#)

## Step 9:

### User Will be created

DATA STORAGE

Clusters

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

We are deploying your changes (current action: configuring MongoDB)

PERSONAL PROJECTS > BLOG\_DEV

**Database Access**

Database Users Custom Roles

User Name	Authentication Method	MongoDB Roles	Resources	Actions
blog_admin	SCRAM	readWriteAnyDatabase@admin	1 Cluster, 0 Data Lakes	<a href="#">EDIT</a> <a href="#">DELETE</a>

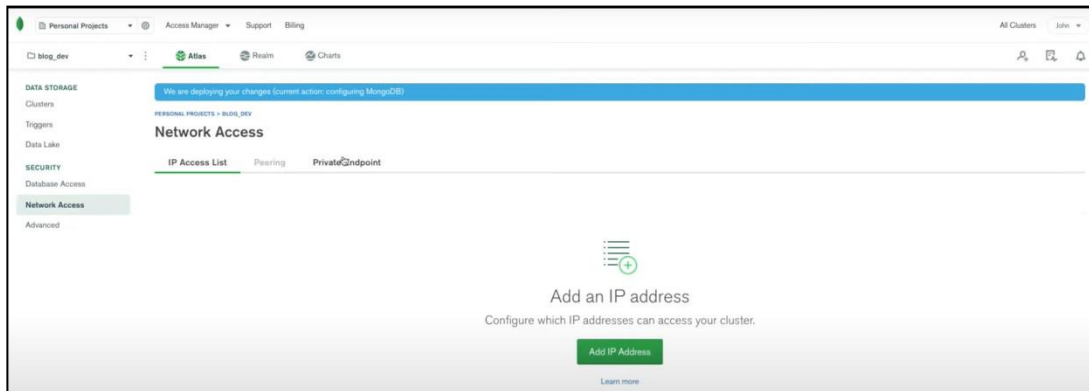
[+ ADD NEW DATABASE USER](#)

**You can Also Edit and Delete User Using Right Side Options.**

## Step 10:

### Create Networks Access for Database

**Go to left side panel click Networks Access options. And then Click the Add IP Address Button.**



### Step 10:

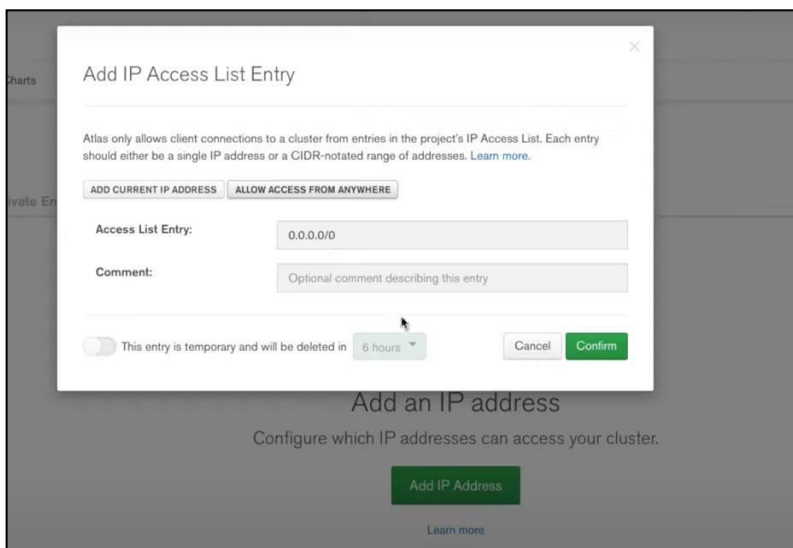
A new popup windows will be open. There are two options are available for IP address

1. Add Current IP Address

These options are more secure and database cluster share with authorize user using generated IP address.

2. Allow Access from Anywhere

These options are not secure and any one can access and use the cluster using generated IP address.



### Step 11:

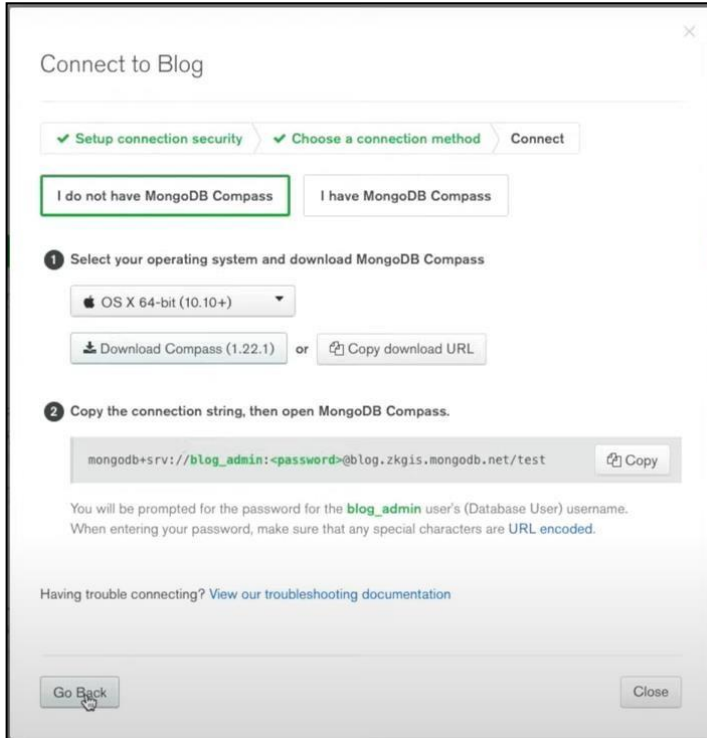
**Connection the cluster with local device choose the connect button.**





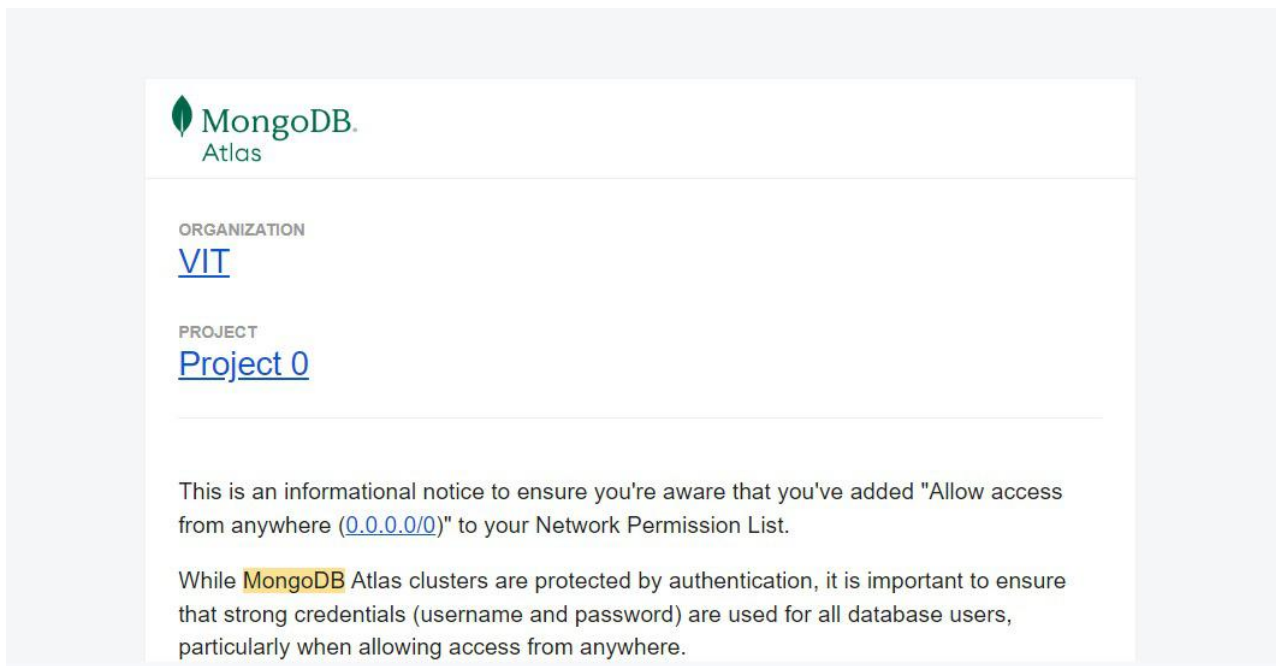


## Follows the Following Steps



The screenshot shows the 'Connect to Blog' dialog box in MongoDB Compass. It has a close button (X) in the top right corner. The dialog is divided into three steps: 'Setup connection security' (completed), 'Choose a connection method' (active), and 'Connect'. Under 'Choose a connection method', there are two buttons: 'I do not have MongoDB Compass' (highlighted with a green border) and 'I have MongoDB Compass'. Step 1, 'Select your operating system and download MongoDB Compass', shows a dropdown menu for 'OS X 64-bit (10.10+)' and two buttons: 'Download Compass (1.22.1)' and 'Copy download URL'. Step 2, 'Copy the connection string, then open MongoDB Compass.', shows a text field with the connection string 'mongodb+srv://blog\_admin:<password>@blog.zkgis.mongodb.net/test' and a 'Copy' button. Below the text field, there is a note: 'You will be prompted for the password for the blog\_admin user's (Database User) username. When entering your password, make sure that any special characters are URL encoded.' At the bottom, there is a link 'Having trouble connecting? View our troubleshooting documentation' and two buttons: 'Go Back' and 'Close'.

An email with the details will be received on the account you signed up with



The screenshot shows the MongoDB Atlas organization and project settings page. The page has a light blue header with the MongoDB Atlas logo. Below the header, there is a section for 'ORGANIZATION' with the name 'VIT' and a 'PROJECT' section with the name 'Project 0'. Below these sections, there is a message: 'This is an informational notice to ensure you're aware that you've added "Allow access from anywhere (0.0.0.0/0)" to your Network Permission List.' and a note: 'While MongoDB Atlas clusters are protected by authentication, it is important to ensure that strong credentials (username and password) are used for all database users, particularly when allowing access from anywhere.'