# Learning Outcome

After completing this module, the student should be able to configure an embedded database with different web pages using MongoDB

To meet the learning outcome, a student has to complete the following activities

1. Create a MongoDB query to display all the trainees who attended course on PHP (2Hrs)
2. Create a MongoDB query to display the 1st batch trainees of PHP (2Hrs)
3. Create a MongoDB query to display the 2nd batch trainees of PHP (2Hrs)
4. Create a MongoDB query to find the course where maximum trainees attended (2Hrs)
5. Create a MongoDB query to find lab wise details of trainees (2Hrs)
6. Create a MongoDB query with course wise details of trainees (2Hrs)
7. Create MongoDB cluster in MongoDB Atlas cloud service (2Hrs)
8. Connect and use MongoDB Atlas on local system (2Hrs)
9. Get familiar with big data and its tools (2Hrs)

# 

# Activity 1

## Aim: Create a MongoDB query to display all the trainees who attended courses on PHP.

**Learning outcome:** Able to configure embedded databases with different web pages using MongoDB.

**Duration:** 2 hours

**List of Hardware/Software requirements:**

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

**Program:**

<?php

//connecting to database

if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017")) echo "Database Connected<br><br>";

$filter=[‘lab’=> ‘PHP’]; //no filters

$option=[]; //options to eliminate lab column from result

$read=new MongoDB\Driver\Query($filter, $option); //creating query

$single\_user=$con->executeQuery("mydb3.mycol", $read); //executing query

//creating table view without column for lab

echo "<table border=3 cellspacing=5 cellpadding=7><thead><th>ID<th>Name<th>Certificate No.<th>Course<th>Start Date<th>End Date</thead>";

foreach($single\_user as $user){ //parsing results in loop echo "<tr>";

echo

"<td>".$user->\_id."<td>".$user->name."<td>".$user->certNo."<td>".$user->course."<td>".$user

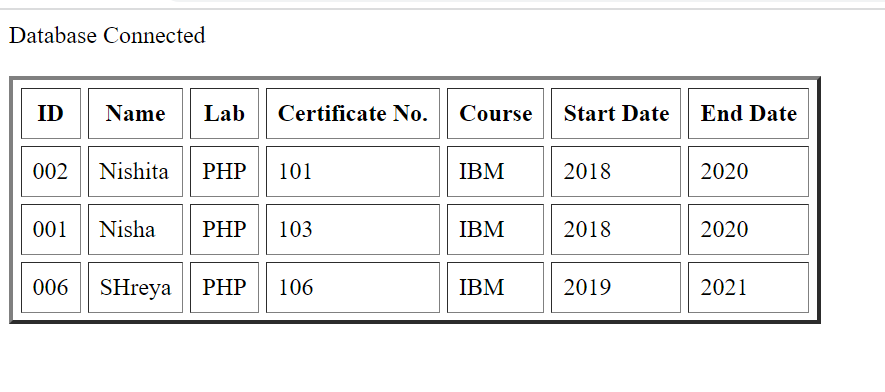
->start."<td>".$user->end;

} end loop

echo "</table>"; //end table

?>

**Output:**

****

# Activity 2

## Aim: Create a MongoDB query to display the 1st batch trainees of PHP

**Learning outcome:** Able to configure embedded databases with different web pages using MongoDB.

**Duration:** 2 hours

**List of Hardware/Software requirements:**

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

**Program:**

<?php

//connecting to database

if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017")) echo "Database Connected<br><br>";

$filter=[‘lab’=> ‘PHP’, ‘start’=> ‘2018’]; //no filters

$option=[]; //options to eliminate lab column from result

$read=new MongoDB\Driver\Query($filter, $option); //creating query

$single\_user=$con->executeQuery("mydb3.mycol", $read); //executing query

//creating table view without column for lab

echo "<table border=3 cellspacing=5 cellpadding=7><thead><th>ID<th>Name<th>Certificate No.<th>Course<th>Start Date<th>End Date</thead>";

foreach($single\_user as $user){ //parsing results in loop echo "<tr>";

echo

"<td>".$user->\_id."<td>".$user->name."<td>".$user->certNo."<td>".$user->course."<td>".$user

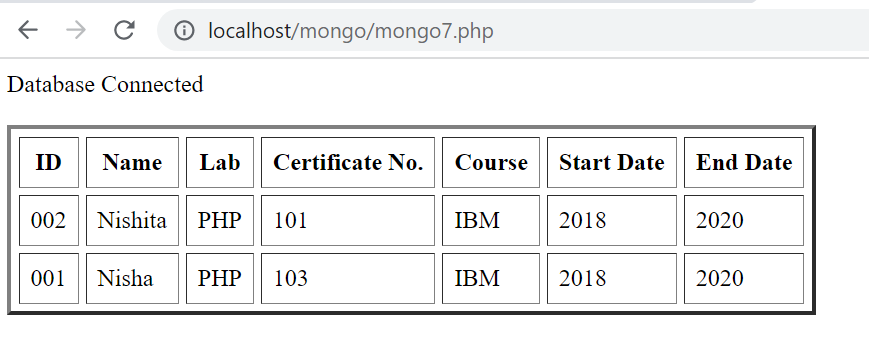
->start."<td>".$user->end;

} end loop

echo "</table>"; //end table

?>

**Output:**

****

# Activity 3

## Aim: Create a MongoDB query to display the 2nd batch trainees of PHP

**Learning outcome:** Able to configure embedded databases with different web pages using MongoDB.

**Duration:** 2 hours

**List of Hardware/Software requirements:**

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

**Program:**

<?php

//connecting to database

if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017")) echo "Database Connected<br><br>";

$filter=[‘lab’=> ‘PHP’, ‘start’=> ‘2019’]; //no filters

$option=[]; //options to eliminate lab column from result

$read=new MongoDB\Driver\Query($filter, $option); //creating query

$single\_user=$con->executeQuery("mydb3.mycol", $read); //executing query

//creating table view without column for lab

echo "<table border=3 cellspacing=5 cellpadding=7><thead><th>ID<th>Name<th>Certificate No.<th>Course<th>Start Date<th>End Date</thead>";

foreach($single\_user as $user){ //parsing results in loop echo "<tr>";

echo

"<td>".$user->\_id."<td>".$user->name."<td>".$user->certNo."<td>".$user->course."<td>".$user

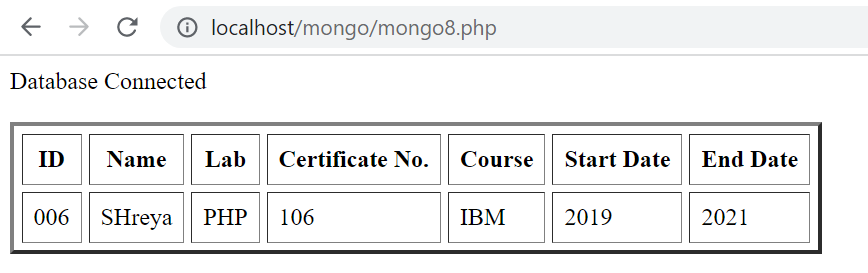
->start."<td>".$user->end;

} end loop

echo "</table>"; //end table

?>

**Output:**



**References** :

● HTML Introduction - https://www.w3schools.com/

● https://www.tutorialspoint.com/mongodb/mongodb\_php.htm

# Activity 4

## Aim: Create a MongoDB query to find the course where maximum trainees attended

**Learning outcome:** Able to configure embedded databases with different web pages using MongoDB.

**Duration:** 2 hours

**List of Hardware/Software requirements:**

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

**Program:**

<?php

//connecting to database

if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))

echo "Database Connected<br><br>";

//no filter or options

$filter=[ ];

$option=[ ];

$ibm=0; //counetr for IBM course

$ef=0; //counter for EF course

$read=new MongoDB\Driver\Query($filter, $option); //create query

$single\_user=$con->executeQuery("mydb3.mycol", $read); //execute query

foreach($single\_user as $user){ //loop for parsing all data

if($user->course=='IBM')

$ibm++; //counter update if course is IBM

else $ef++; //counter update if course is EF

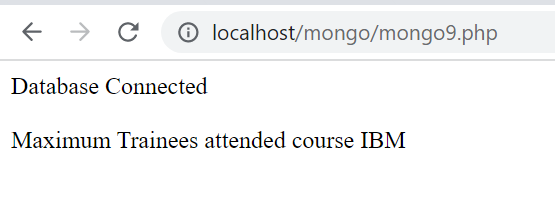
}

if($ibm>$ef) echo('Maximum Trainees attended course IBM'); //display if IBM is max

else echo('Maximum Trainees attended course EF'); //display if EF is max

?>

**Output:**



# Activity 5

**Aim:** Create a MongoDB query to find lab wise details of trainees.

**Learning outcome:** Able to configure embedded databases with different web pages using MongoDB.

**Duration:** 2 hours

**List of Hardware/Software requirements:**

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

**Program:**

<?php

//connecting to database

if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))

echo "Database Connected<br><br>";

$filter=[ ]; //no filters

$option=[ 'sort'=>['lab'=>1] ]; //sorting options on course name

$read=new MongoDB\Driver\Query($filter, $option); //create query

$single\_user=$con->executeQuery("mydb3.mycol", $read); //execute query

//create table view

echo "<table border=3 cellspacing=5

cellpadding=7><thead><th>ID<th>Name<th>Lab<th>Certificate No.<th>Course<th>Start

Date<th>End Date</thead>";

foreach($single\_user as $user){ //parsing results though loop

echo "<tr>";

echo

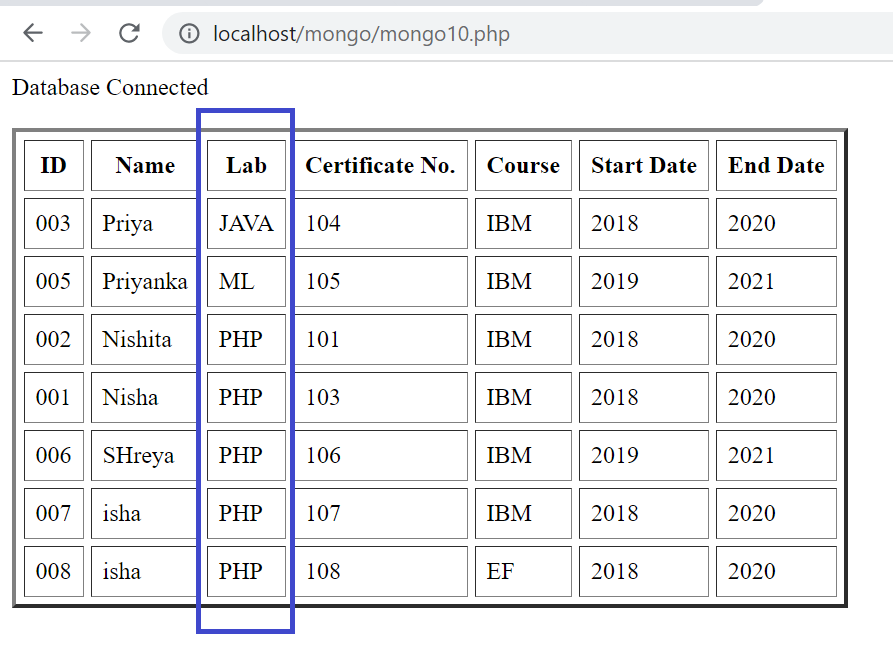
"<td>".$user->\_id."<td>".$user->name."<td>".$user->lab."<td>".$user->certNo."<td>".$user->course."<td>".$user->start."<td>".$user->end;

} //end loop

echo "</table>"; //end table

?>

**Output:**



# Activity 6

**Aim:** Create a MongoDB query to find Course wise details of trainees.

**Learning outcome:** Able to configure embedded databases with different web pages using MongoDB.

**Duration:** 2 hours

**List of Hardware/Software requirements:**

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4

**Program:**

<?php

//connecting to database

if($con=new MongoDB\Driver\Manager("mongodb://localhost:27017"))

echo "Database Connected<br><br>";

$filter=[ ]; //no filters

$option=[ 'sort'=>['course'=>1] ]; //sorting options on course name

$read=new MongoDB\Driver\Query($filter, $option); //create query

$single\_user=$con->executeQuery("mydb3.mycol", $read); //execute query

//create table view

echo "<table border=3 cellspacing=5

cellpadding=7><thead><th>ID<th>Name<th>Lab<th>Certificate No.<th>Course<th>Start

Date<th>End Date</thead>";

foreach($single\_user as $user){ //parsing results though loop

echo "<tr>";

echo

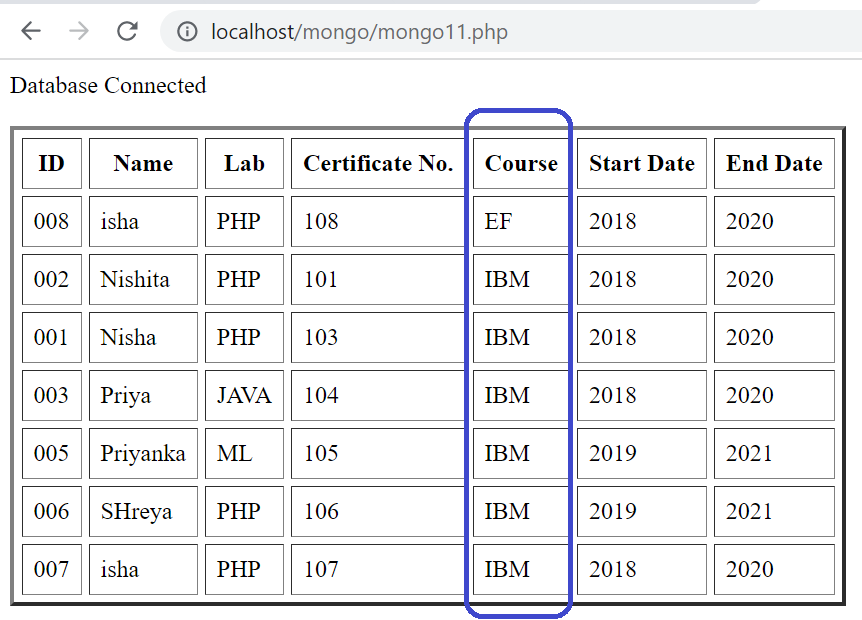
"<td>".$user->\_id."<td>".$user->name."<td>".$user->lab."<td>".$user->certNo."<td>".$user->course."<td>".$user->start."<td>".$user->end;

} //end loop

echo "</table>"; //end table

?>

**Output:**



# Activity 7

**Aim:** Create MongoDB cluster in MongoDB Atlas cloud service

**Learning outcome:** Able to create MongoDB cluster on MongoDB Atlas cloud platform.

**Duration:** 2 hours

**List of Hardware/Software requirements:**

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS / Windows
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4
5. MongoDB Atlas cloud account

**Procedure/ code:**

## [Set up a free MongoDB Atlas cluster](https://studio3t.com/knowledge-base/articles/mongodb-atlas-tutorial/#set-up-a-free-mongodb-atlas-cluster)

1. Go to the [MongoDB Atlas landing page](https://www.mongodb.com/cloud/atlas/register).

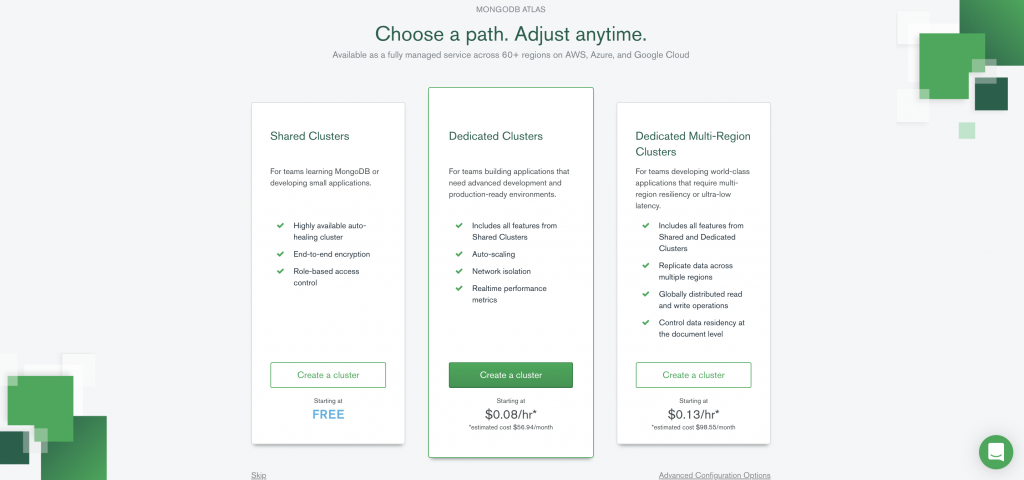
2. Fill in the required information (email address, first name, last name, and password).

3. Click the terms of service and privacy policy links, which should open on a new tab.  
If you want to continue with the registration, select the **I agree to the terms of service and privacy policy** check box.

5. Click the **Get started free** at the bottom of the form.

6. The website will ask to choose a cluster. Choose **Starter Clusters** and click on **Create a cluster**.

You should also receive a welcome email, which confirms that you’re registered with MongoDB Atlas and includes a link for logging onto the service.

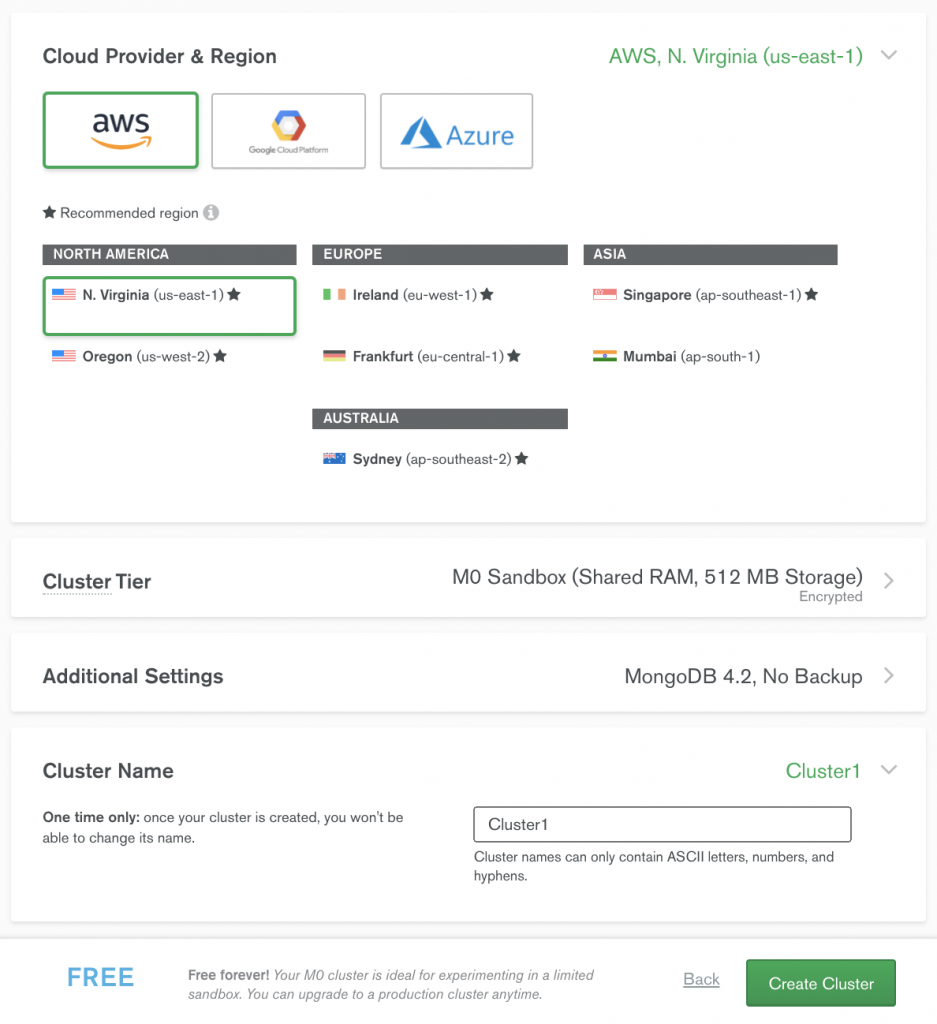


7. In the **Cloud Provider & Region** section, the **aws** option should be selected as the default provider, but you can select any provider. All three platforms support the free tier.

8. Beneath the list of providers, select a region.

9. Expand the **Cluster Tier** section and ensure that **M0 Sandbox** is selected. This is the free M0 service level.

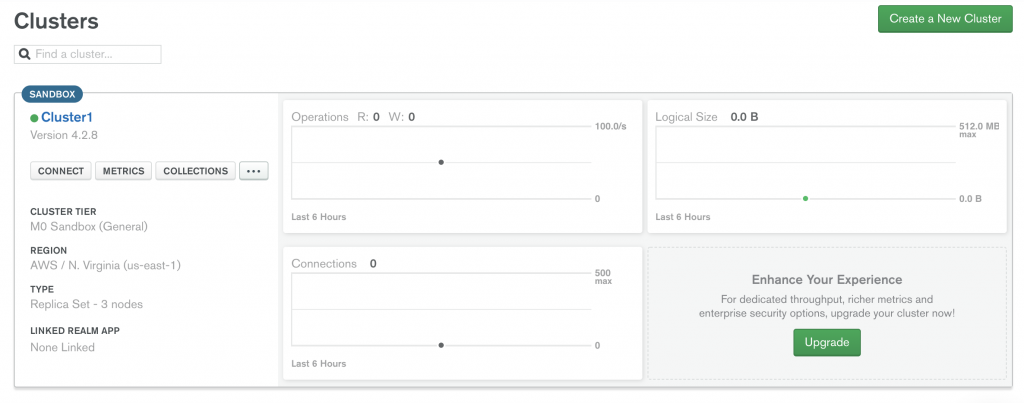
10. Expand the **Cluster Name** section and type **Cluster1** in the text box.



11. Click the **Create Cluster** button at the bottom of the web page.

12. You should then receive a message stating that your cluster is being created.  
When the process is complete, you’ll be taken to the Clusters page, which includes a listing for your new cluster, as shown in the following figure.

**Output:**



The cluster information includes the service level, cloud provider, and region, along with details about operations, connections and logical size, all of which currently show zero amounts.

**Reference:**

[**https://studio3t.com/knowledge-base/articles/mongodb-atlas-tutorial/**](https://studio3t.com/knowledge-base/articles/mongodb-atlas-tutorial/)

[**https://cloud.mongodb.com/v2/60a4bd5bf765a20087add3a7#clusters/edit?from=ctaClusterHeader**](https://cloud.mongodb.com/v2/60a4bd5bf765a20087add3a7#clusters/edit?from=ctaClusterHeader)

# Activity 8

**Aim:** Connect and use MongoDB Atlas on local system

**Learning outcome:** configure MongoDB Atlas to connect to the cluster you created.

**Duration:** 2 hours

**List of Hardware/Software requirements:**

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS / Windows
2. MongoDB 4.2 Community Edition
3. XAMPP- PHP 7
4. PHP-MongoDB driver 1.7.4
5. MongoDB Atlas cloud account

**Procedure/ code:**

## [Configure IP address and connection string](https://studio3t.com/knowledge-base/articles/mongodb-atlas-tutorial/#configure-ip-address-and-connection-string)

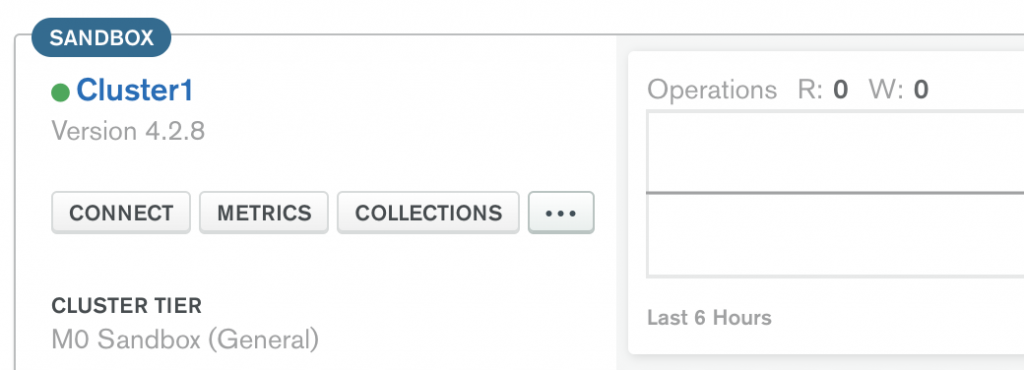
you’ll configure MongoDB Atlas to connect to the cluster you created.

**For this, you will need the IP address of the device that will connect to the service.**

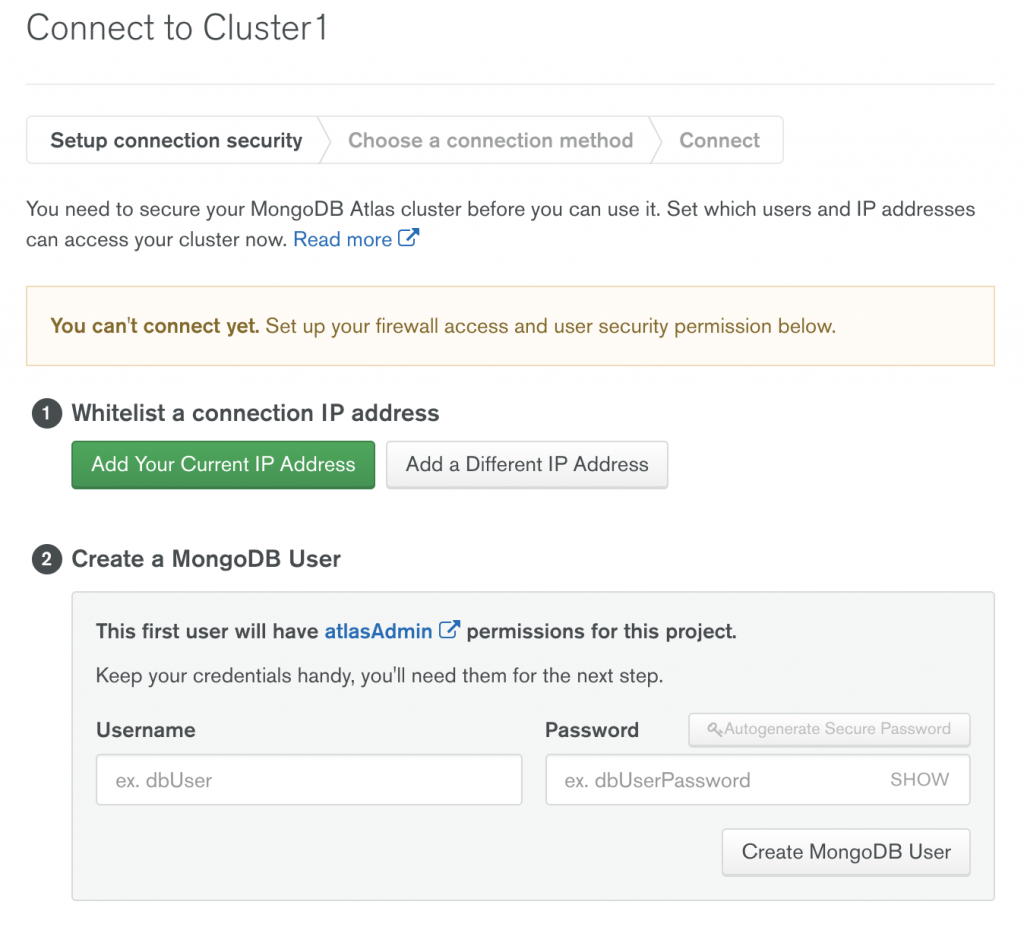
If you plan to connect to MongoDB Atlas on the same device where you’re setting up the service, MongoDB Atlas can find the local IP address automatically.

As part of this exercise, you’ll also set up an administrator account for accessing the cluster.

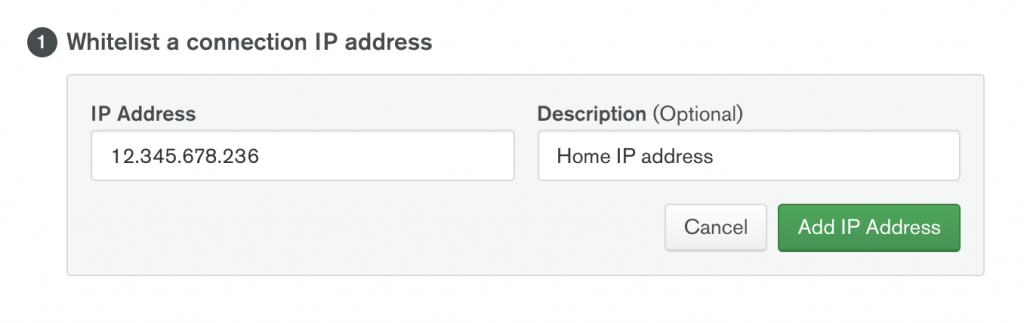
1. In the **Cluster1** section of the **Clusters** page, click the **CONNECT** button in the left pane.



The **Connect to Cluster1** dialog box appears, showing the two steps you must take to configure your connection.



1. For security reasons, MongoDB Atlas blocks all outside connections by default. In order to connect, you must first whitelist your IP address. In the Step 1 section, do one of the following:



* If you plan to connect from the computer you’re currently using, click **Add Your Current IP Address** and then click **Add IP Address**.
* If you plan to connect from a different computer, click **Add a Different IP Address**, type the IP address, and then click **Add IP Address**. You can edit the IP address, add IP addresses, or delete them at any time in case you change devices.

3. In the Step 2 section, type **admin** in the **Username** text box (or whatever name you want to use), and then type a password in the **Password** text box.

To make it easier to connect to MongoDB Atlas from Studio 3T, your password should include only alphanumeric characters, that is, letters and numbers only with no special characters.

If you use special characters, you will need to encode them when creating a connection string for accessing the MongoDB service.

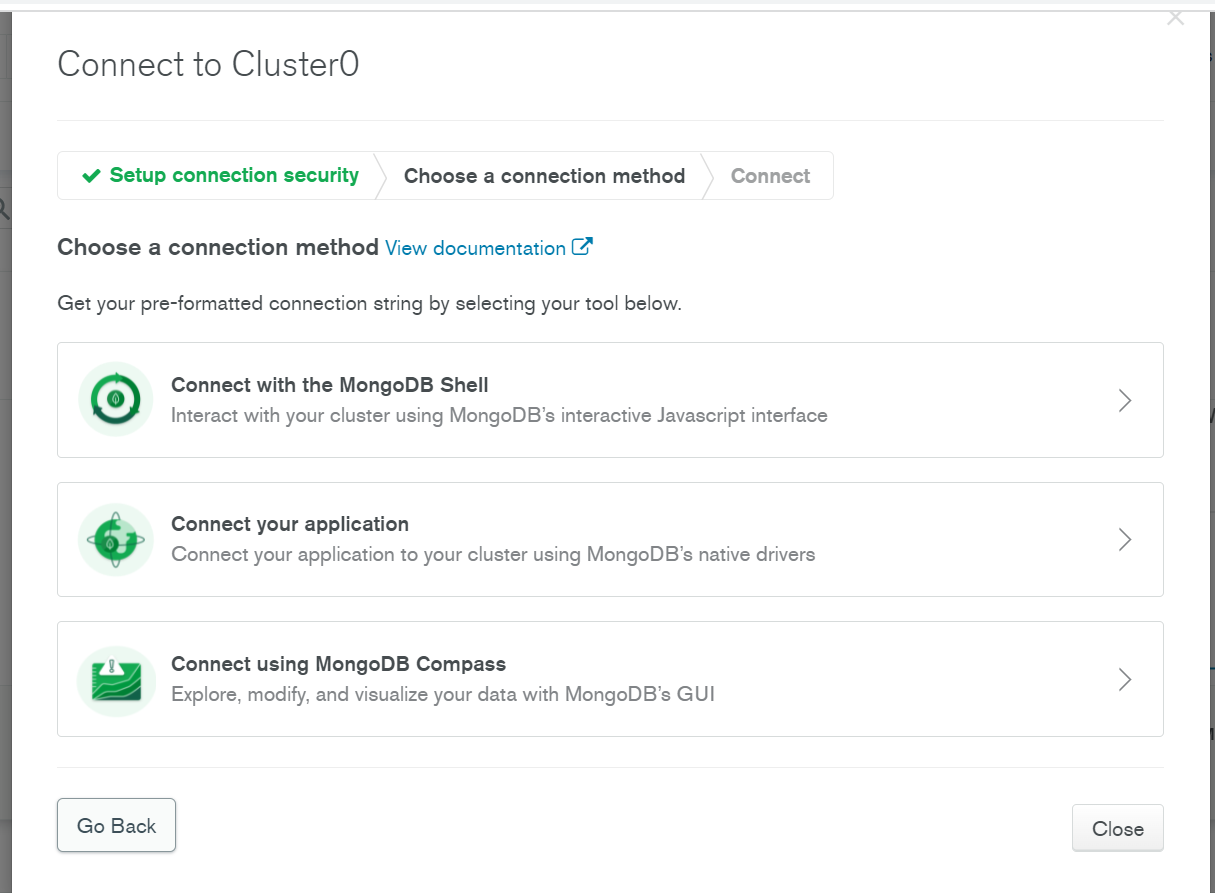
**Note:** To generate a password automatically, click the **Autogenerate Secure Password** button. A password will be generated that includes only alphanumeric characters. Be sure to save the password somewhere secure.

4. Click the **Create MongoDB User** button.

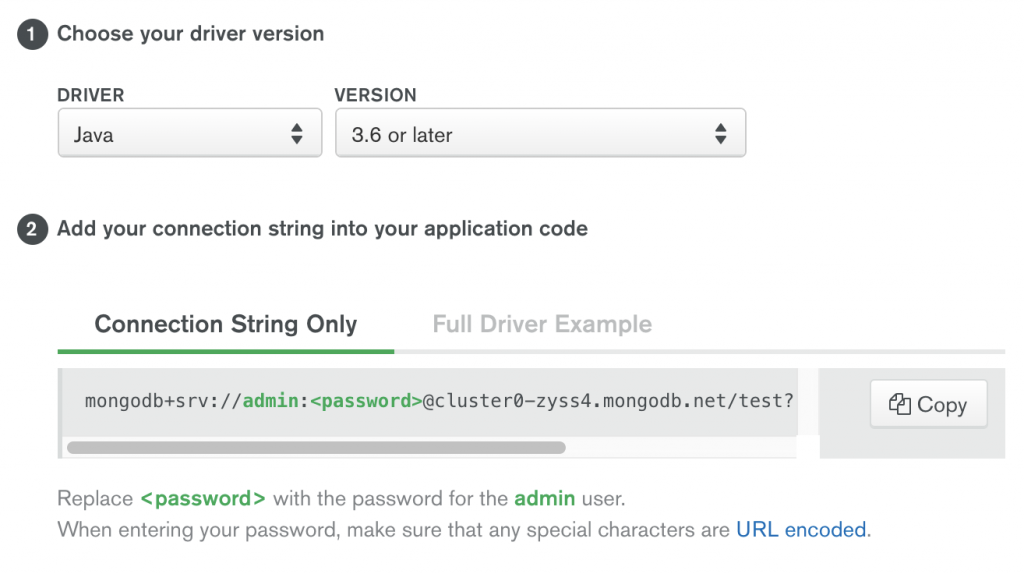
5. The next task is to generate the connection string or Uniform Resource Identifier (URI).

To start this process, click the **Choose a connection method** button. The Connect to Cluster1 dialog box reappears, providing a different set of options.

1. Click **Connect Your Application**. You’ll again be presented with two steps.



7. In the Step 1 section, select **Java** from the **DRIVER** drop-down list, and select **3.6 or later** from the **VERSION** drop-down list.

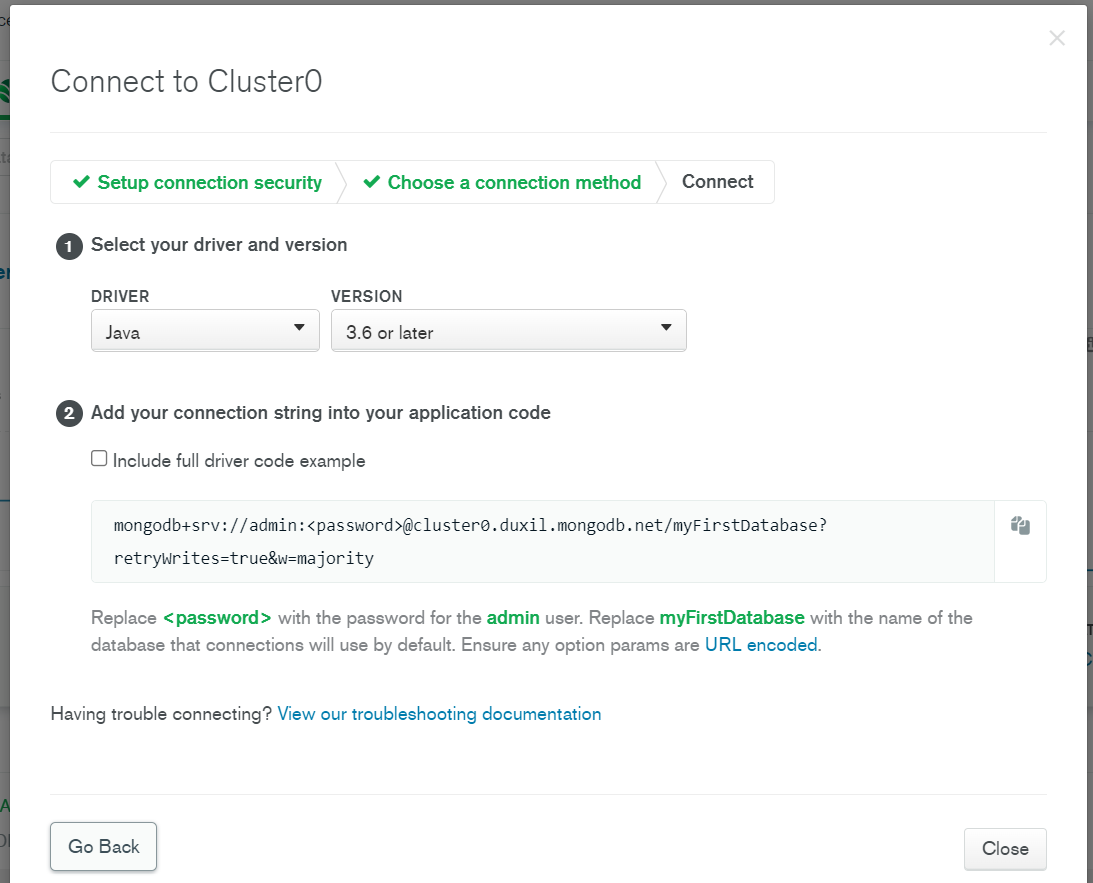


8. In the Step 2 section, click **Copy** to copy the connection string to your clipboard, and then paste the connection string to a safe location.

When you use the connection string or URI, you must replace the placeholder with the password you created for the administrator account. **Don’t forget to remove the <> as well.**

9.Click **Close** to close the Connect to Cluster1 dialog box, and then sign out of the MongoDB Atlas service.

**Output:**

****

**Reference:**

<https://studio3t.com/knowledge-base/articles/mongodb-atlas-tutorial/>

# Activity 9

**Aim:** Get familiar with big data and its tools

**Learning outcome:** Get familiar with big data and its tools

**Duration:** 2 hours

**List of Hardware/Software requirements:**

1. Laptop/Computer with Linux OS - Ubuntu 18.04 LTS / Windows
2. Big Data Tools

Procedure/

**Big data** is simply too large and complex data that cannot be dealt with using traditional data processing methods.

Big Data requires a set of tools and techniques for analysis to gain insights from it.

There are a number of big data tools available in the market such as Hadoop which helps in storing and processing large data, Spark helps in-memory calculation, Storm helps in faster processing of unbounded data, Apache Cassandra provides high availability and scalability of a database, MongoDB provides cross-platform capabilities, so there are different functions of every Big Data tool.

Analyzing and processing Big Data is not an easy task. Big Data is one big problem and to deal with it you need a set of great big data tools that will not only solve this problem but also help you in producing substantial results.

## What are the best Big Data Tools?

Here is the list of top 10 big data tools –

* Apache Hadoop
* Apache Spark
* Flink
* Apache Storm
* Apache Cassandra
* MongoDB
* Kafka
* Tableau
* RapidMiner
* R Programming

**Big Data** is an essential part of almost every organization these days and to get significant results through Big Data Analytics a set of tools is needed at each phase of data processing and analysis.

There are a few factors to be considered while opting for the set of tools i.e., the size of the datasets, pricing of the tool, kind of analysis to be done, and many more.

With the exponential growth of Big Data, the market is also flooded with its various tools. These tools used in big data help in bringing out better cost efficiency and thus increases the speed of analysis.

**Apache Hadoop**

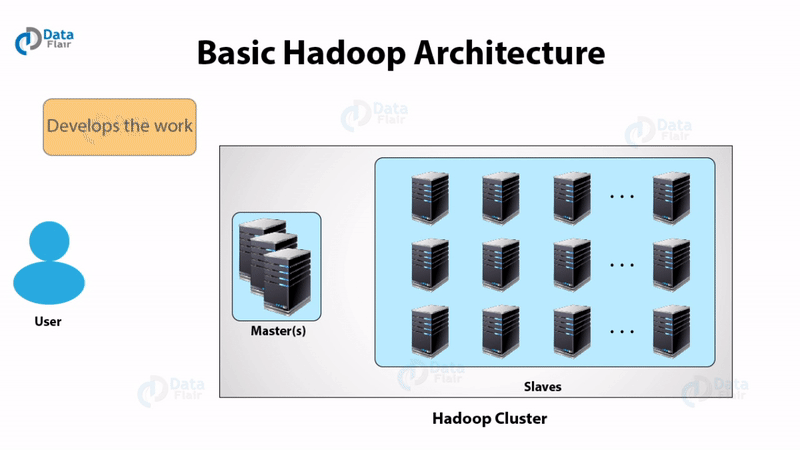
**Apache Hadoop** is one of the most popularly used tools in the Big Data industry.

Hadoop is an open-source framework from Apache and runs on commodity hardware. It is used to store process and analyze Big Data.

Hadoop is written in Java. Apache Hadoop enables parallel processing of data as it works on multiple machines simultaneously. It uses clustered architecture. A Cluster is a group of systems that are connected via LAN.

It consists of 3 parts-

1. **Hadoop Distributed File System (HDFS) –** It is the storage layer of Hadoop.
2. **Map-Reduce –** It is the data processing layer of Hadoop.
3. **YARN –** It is the resource management layer of Hadoop.



**Image Source:** [**https://data-flair.training/blogs/top-big-data-tools/**](https://data-flair.training/blogs/top-big-data-tools/)

Limitation of Apache Hadoop:

* Hadoop does not support real-time processing. It only supports batch processing.
* Hadoop cannot do in-memory calculations.

### **Apache Spark**

**Apache Spark**can be considered as the successor of Hadoop as it overcomes the drawbacks of it. Spark, unlike Hadoop, supports both real-time as well as batch processing. It is a general-purpose clustering system.

It also supports in-memory calculations, which makes it 100 times faster than Hadoop. This is made possible by reducing the number of read/write operations into the disk.

It provides more flexibility and versatility as compared to Hadoop since it works with different data stores such as HDFS, OpenStack and Apache Cassandra.

It offers high-level APIs in Java, Python, Scala and R. Spark also offers a substantial set of high-level tools including Spark SQL for structured data processing, MLlib for machine learning, GraphX for graph data set processing, and Spark Streaming. It also consists of 80 high-level operators for efficient query execution.

**Apache Storm**

Apache Storm is an open-source big data tool, distributed real-time and fault-tolerant processing system. It efficiently processes unbounded streams of data.

By unbounded streams, we refer to the data that is ever-growing and has a beginning but no defined end.

The biggest advantage of Apache Storm is that it can be used with any of the programming languages and it further supports JSON based protocols.

The processing speed of Storm is very high. It is easily scalable and also fault-tolerant. It is much easier to use.

On the other hand, it guarantees the processing of each data set. It’s processing speed is rapid and a standard observed was as high as a million tuples processed per second on each node.

**Apache Cassandra**

*Apache Cassandra* is a distributed database that provides high availability and scalability without compromising performance efficiency. It is one of the best big data tools that can accommodate all types of data sets namely structured, semi-structured, and unstructured.

It is the perfect platform for mission-critical data with no single point of failure and provides fault tolerance on both commodity hardware and cloud infrastructure.

Cassandra works quite efficiently under heavy loads. It does not follow master-slave architecture so all nodes have the same role. Apache Cassandra supports the ACID (Atomicity, Consistency, Isolation, and Durability) properties.

**MongoDB**

**MongoDB** is an open-source data analytics tool, NoSQL database that provides cross-platform capabilities. It is exemplary for a business that needs fast-moving and real-time data for taking decisions.

MongoDB is perfect for those who want data-driven solutions. It is user-friendly as it offers easier installation and maintenance. MongoDB is reliable as well as cost-effective.

It is written in C, C++, and JavaScript. It is one of the most popular databases for Big Data as it facilitates the management of unstructured data or the data that changes frequently.

MongoDB uses dynamic schemas. Hence, you can prepare data quickly. This allows in reducing the overall cost. It executes on MEAN software stack, NET applications and, Java platform. It is also flexible in cloud infrastructure.

But a certain downfall in the processing speed has been noticed for some use-cases.

**Apache Flink**

Apache Flink is an Open-source data analytics tool distributed processing framework for bounded and unbounded data streams. It is written in Java and Scala. It provides high accuracy results even for late-arriving data.

Flink is a stateful and fault-tolerant i.e. it has the ability to recover from faults easily. It provides high-performance efficiency at a large scale, performing on thousands of nodes.

It gives a low-latency, high throughput streaming engine and supports event time and state management.

**Kafka**

**Apache Kafka** is an open-source platform that was created by LinkedIn in the year 2011.

Apache Kafka is a distributed event processing or streaming platform which provides high throughput to the systems. It is efficient enough to handle trillions of events a day. It is a streaming platform that is highly scalable and also provides great fault tolerance.

The streaming process includes publishing and subscribing to streams of records alike to the messaging systems, storing these records durably, and then processing these records. These records are stored in groups called topics.

Apache Kafka offers high-speed streaming and guarantees zero downtime.

**Tableau**

**Tableau** is one of the best data visualization and software solution tools in the Business Intelligence industry. It’s a tool that unleashes the power of your data.

It turns your raw data into valuable insights and enhancing the decision-making process of the businesses.

Tableau offers a rapid data analysis process and resulted in visualizations are in the form of interactive dashboards and worksheets.

It works in synchronization with other **Big Data tools** such as Hadoop.

**Tableau** offered the capabilities of data blending are best in the market. It provides an efficient real-time analysis.

**Tableau** is not only bound to the technology industry but is a crucial part of some other industries as well. This software doesn’t require any technical or programming skills to operate.

**RapidMiner**

RapidMiner is a cross-platform tool that provides a robust environment for Data Science, Machine Learning and Data Analytics procedures. It is an integrated platform for the complete Data Science lifecycle starting from data prep to machine learning to predictive model deployment.

It offers various licenses for small, medium, and large proprietary editions. Apparently, it also offers a free edition that permits only 1 logical processor and up to 10,000 data rows.

RapidMiner is an open-source tool that is written in java. RapidMiner offers high efficiency even when integrated with APIs and cloud services. It provides some robust Data Science tools and algorithms.

**R Programming**

R is an open-source programming language and is one of the most comprehensive statistical analysis languages.

It is a multi-paradigm programming language that offers a dynamic development environment. As it is an open-source project and thousands of people have contributed to the development of the R.

R is written in C and Fortran. It is one of the most widely used statistical analysis tools as it provides a vast package ecosystem.

It facilitates the efficient performance of different statistical operations and helps in generating the results of data analysis in graphical as well as text format. The graphics and charting benefits it provides are unmatchable.

Reference:

<https://data-flair.training/blogs/top-big-data-tools/>