

Introduction to MongoDB

Objectives

- ❑ By the end of this session, you should be able to:
 - Database Management System
 - Managing MongoDB
 - MongoDB Atlas
 - MongoDB Clusters
 - MongoDB Documents
 - MongoDB Collections
 - MongoDB Databases

Introduction to MongoDB

- ❑ MongoDB is a document-oriented database. This means that it doesn't use tables and rows to store its data, but instead *collections* of JSON-like *documents*. These documents support embedded fields, so related data can be stored within them.
- ❑ MongoDB is also a schema-less database, so we don't need to specify the number or type of columns before inserting our data.
- ❑ MongoDB is a popular NoSQL database that can store both structured and unstructured data. Founded in 2007 by Kevin P. Ryan, Dwight Merriman, and Eliot Horowitz in New York, the organization was initially called 10gen and was later renamed MongoDB—a word inspired by the term humongous.

Managing MongoDB

- ❑ MongoDB provides the user with two options. Based on your requirements, you can either install it on your system and manage the database yourself or utilize the **Database as a Service (DBaaS)** option offered by MongoDB (**Atlas**).
- ❑ **Self-Managed**
 - MongoDB is available to be downloaded and installed on your machines. The machine can be a workstation, a server, a virtual machine in a data center, or on the cloud. You can install MongoDB as standalone, a replica set, or sharded clusters. All these deployments are possible with both the Community and Enterprise Editions. Each deployment has its advantages and associated complexity. A self-managed database can be useful for scenarios where you either want more granular control of your database or you just want to learn database management and operations.
- ❑ **Managed Service: Database as a Service**
 - A managed service is the concept of outsourcing some processes, functions, or deployments to a vendor. DBaaS is a term generally used for databases outsourced to an external vendor. A managed service enforces a shared responsibility model. The provider of the service manages the infrastructure, that is, the installation, deployment, failover, scalability, disk space, monitoring, and so on. You can manage the data and the settings for security, performance, and tuning. It allows you to save time managing databases and focus on other things, such as application development.

MongoDB Atlas

- ❑ MongoDB Atlas is the DBaaS offering from MongoDB Inc. It allows you to provision a database on the cloud as a service, which can be used for your applications from anywhere. Atlas uses cloud infrastructures from different cloud vendors. You can choose the cloud vendor on which you want to deploy your database. Like any other
- ❑ managed service, you get the benefits of highly available secured environments with
- ❑ low or no maintenance needed.

MongoDB Atlas (contd.)

Benefits:

- Simple Setup
- Guaranteed Availability
- Global Presence
- Optimal Performance
- Highly Secured
- Automated Backups

MongoDB Clusters

- ❑ A MongoDB cluster is the term used for a database replica set or shared deployments in MongoDB Atlas. A cluster is a distributed set of servers used for data storage and retrieval. A MongoDB cluster, at the minimum level, is a three-node replica set.
- ❑ In a sharded environment, a single cluster may contain hundreds of nodes/servers containing different replica sets with each replica set comprised of at least three nodes/servers.

MongoDB Documents

- ❑ MongoDB stores data records in documents.
- ❑ A document is a collection of field names and values, structured in a JavaScript Object Notation (JSON)-like format.
- ❑ JSON is an easy-to-understand key-value pair format to describe data.
- ❑ The documents in MongoDB are stored as an extension of the JSON type, which is called BSON (Binary JSON). It is a binary-encoded serialization of JSON-like documents.
- ❑ BSON is designed to be more efficient in space than standard JSON.
- ❑ BSON also contains extensions that allow the representation of data types that cannot be represented in JSON.

MongoDB Documents (contd.)

- ❑ MongoDB documents contain field and value pairs and follow a basic structure, as follows:

```
{  
  "firstFieldName": firstFieldValue, "secondFieldName": secondFieldValue,  
  ...  
  "nthFieldName": nthFieldValue  
}
```

- ❑ The following is another example with some fields and date types from BSON:

```
{  
  "_id" : ObjectId("5da26553fb4ef99de45a6139"), "name" : "Roxana",  
  "dateOfBirth" : new Date("Dec 25, 2007"), "placeOfBirth" : "Brisbane",  
  "profession" : "Student"  
}
```

MongoDB Collections

- ❑ In MongoDB, documents are stored in collections.
- ❑ Collections are analogous to tables in relational databases. You need to use the collection name in your queries for operations such as insert, retrieve, delete, and so on.

MongoDB Databases

- ❑ A database is a container for collections grouped together.
- ❑ Each database has several files on the filesystem that contain database metadata and the actual data stored in collections.
- ❑ MongoDB allows you to have multiple databases, and each of these databases can have various collections.
- ❑ In turn, each of these collections can have numerous documents.
- ❑ This is illustrated in the figure on Next slide, which shows an events database that contains collections for different event-related fields, such as *Person*, *Location*, and *Events*; these, in turn, contain various documents with all the granular data:

MongoDB Databases (contd.)

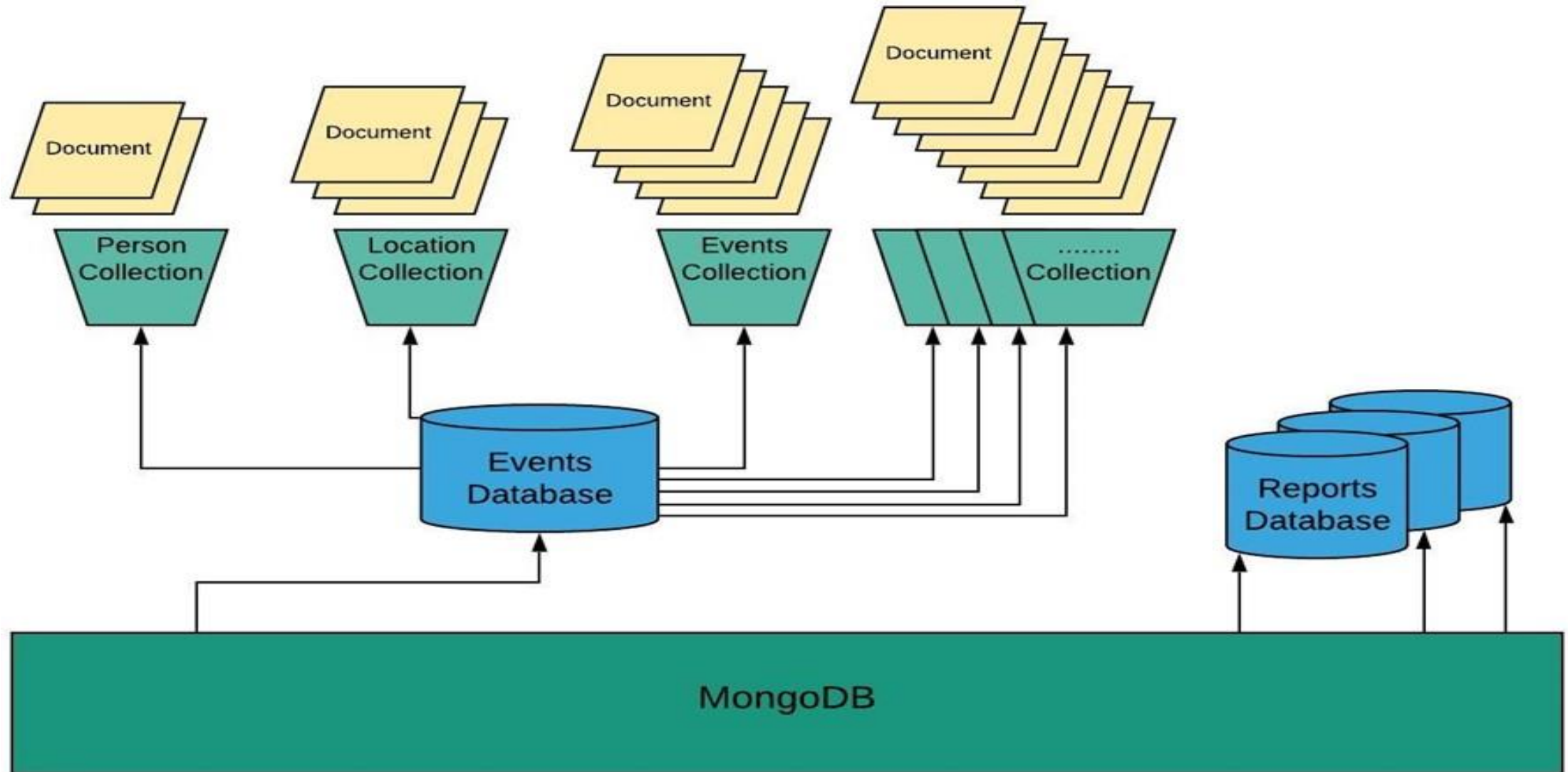


Figure 1.38: Pictorial representation of a MongoDB database

Summary

- ❑ In this session, you learned about:
 - We began this chapter by covering the fundamentals of data, databases, RDBMS, and NoSQL databases.
 - You learned that MongoDB can be used as self-managed or as DbaaS, set up your account in MongoDB Atlas, and reviewed MongoDB deployment on different cloud platforms and how to estimate its cost.
 - We concluded the chapter with the MongoDB structure and its basic components, such as databases, collections, and documents.
- ❑ In the next chapter, you will utilize these concepts to explore MongoDB components and its data model.