

LINKS

<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/> - MongoDB Community Edition

<https://www.mongodb.com/try/download/shell> - Shell

Introduction to MongoDB

* NoSQL Database or non-relational database
* Unstructured
* MongoDB != RDBMS
* Supports Horizontal Scalability
* It doesn’t use tables for storing data
* Used to store big data and real-time web application

Advantages of NoSQL

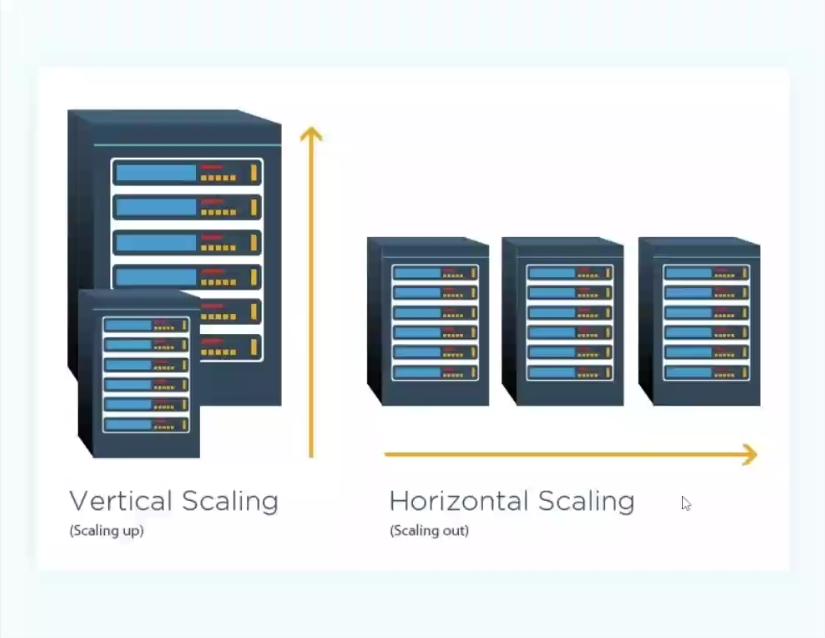
* It supports query language
* It provides fast performance
* It provides horizontal scalability

Vertical Scalability

* Vertical Scaling (Scale Up)  
  Upgrading the **existing machine’s capacity** — more CPU, RAM, or SSD.
* **Example:**  
  Upgrading MongoDB server from 8 GB RAM to 32 GB RAM.
* **Pros:**
  + Simpler to implement
  + No change in application architecture
* **Cons:**
  + Limited by hardware capacity
  + Downtime may be needed for upgrades
* **Applications**
  + Simpler applications or early stages
  + Databases with single-instance support

Horizontal Scalability

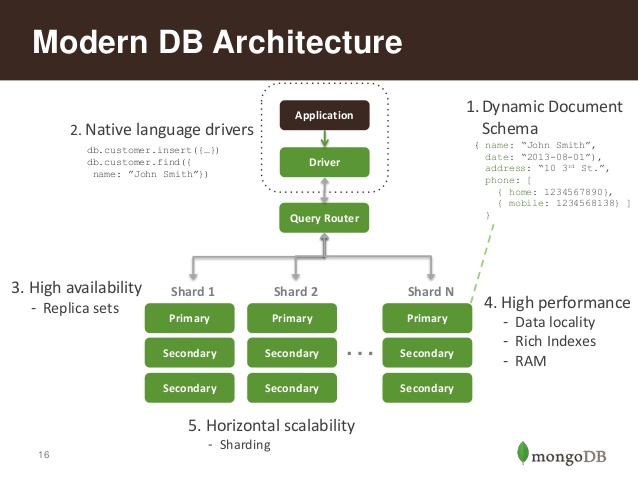
* **Horizontal Scaling (Scale Out)**  
  Adding **more machines/nodes** to your system.()
* **Example:**  
  Instead of running MongoDB on one server, run it on **3 servers**. The data and load are **distributed**.
* **Pros:**
* Fault-tolerant (if one server fails, others still run)
* Better performance for large-scale systems
* Easier to scale indefinitely
* **Cons:**
* More complex (requires load balancing, syncing)
* Higher infrastructure management
* **Applications:**
* Web applications
* Distributed databases (e.g., MongoDB sharding)
* Cloud environments



MongoDB

**MongoDB,**

* Open-source NoSQL document-oriented database
* Stores data in JSON-like BSON documents



**Key Features**

* Schema-less (dynamic schema)
* Horizontal scaling
* Integrated aggregation
* Powerful querying
* High availability via replication

**NoSQL**

* Not Only SQL
* Designed to handle unstructured/semi-structured data
* Stores data in key-value, document, column, or graph formats

**Limitations of Traditional (SQL) Databases**

* Fixed schema
* Complex joins
* Not great with large-scale distributed systems

**Why NoSQL?**

* Flexible schema
* High scalability (horizontal)
* Better performance for Big Data & real-time applications

