DAY-4 (22/5/2025)

Emp ID : 110714877

NAME :Kinnera venkatamma

What is database scalability?

Database scalability is the ability to expand or contract the capacity of system resources in order to support the changing usage of your application. This can refer both to increasing and decreasing usage of the application. Increased usage of your application brings three main challenges to your database server:

The CPU and/or memory becomes overloaded, and the database server either cannot respond to all the request throughput or do so in a reasonable amount of time.

Your database server runs out of storage, and thus cannot store all the data.

Your network interface is overloaded, so it cannot support all the network traffic received.

The first action you might take to address the need for increased capacity is application and database optimization. Examples include optimizing the application code, caching, and appropriately indexing your query patterns. These optimizations increase the efficiency of your application and should bring some relief. However, there comes a point when system resource limits are reached. At this point, you will want to consider scaling your database vertically, horizontally, or both.

What is vertical and horizontal scaling?

Vertical scaling refers to increasing the processing power of a single server or cluster. Both relational and non-relational databases can scale up, but eventually, there will be a limit in terms of maximum processing power and throughput.

Horizontal scaling, also known as scale-out, refers to bringing on additional nodes to share the load.

MongoDB Atlas makes it simple to vertically or horizontally scale up or down as needed. You can even enable auto-scaling so your available resources always meet your needs.

How to increase database performance and make it more scalable

There are a variety of scaling techniques that depend on the database system and what components are used. However, they all use the concept of a node, which is an individual machine storing some or all of the data. A group of nodes that work together is called a cluster.

Resilience:

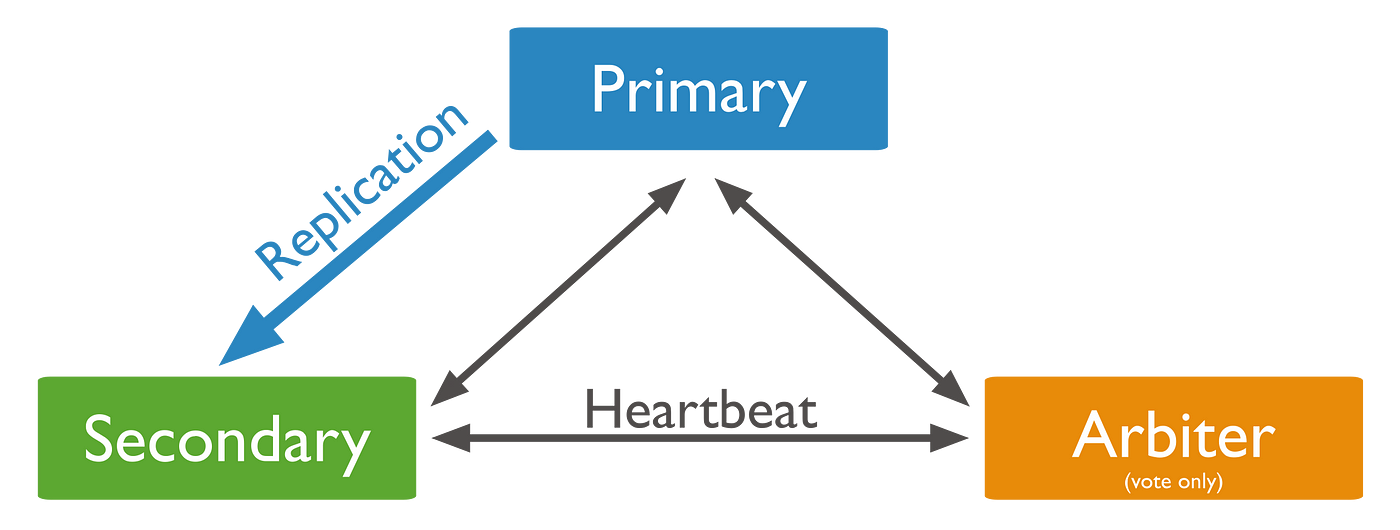
Resilience refers to the ability of a system to maintain its operations and data integrity in the face of disruptions, failures, or unexpected events. This includes both proactive measures to prevent issues and reactive measures to quickly recover from them. MongoDB Atlas, for instance, is designed with resilience in mind, offering features like replica sets, automatic failover, and data replication to ensure high availability and data protection.

Mission -critical :

In MongoDB, "mission-critical" refers to applications or workloads that are essential to the business's overall operations and where downtime or data loss could have significant negative consequences, according to MongoDB. These applications often involve high availability, reliability, and security requirements, according to blog.jsr.wtf.

Replica set:

In MongoDB, a replica set is a group of mongod processes that maintain the same data set, providing redundancy and high availability. It ensures that data is replicated across multiple servers, so if one server fails, the others can take over, preventing data loss and downtime. Replica sets are a fundamental part of MongoDB's production deployments and are the basis for most real-world applications.



Sharding :

In MongoDB, Sharding is a technique used to distribute data across multiple servers (shards) to handle large datasets and high throughput operations. It allows for horizontal scaling, meaning you can add more servers to increase capacity and handle increased demands.

Privacy & Security :

MongoDB privacy and security encompass a range of measures to protect data confidentiality, integrity, and availability. This includes authentication, access control, encryption, and auditing features. MongoDB's data encryption safeguards data both in transit and at rest, preventing unauthorized access, even if physical storage is compromised. Additionally, MongoDB offers role-based access control and auditing mechanisms to manage user permissions and track database activity.

Authentication:

This process verifies that the client connecting to the database is who they claim to be. In MongoDB, this typically involves providing a username and password.

Authorization:

Once the client is authenticated, authorization determines what resources and actions they are allowed to access. This is often controlled through roles and permissions assigned to users within the database.

Authentication & Encryption :

In MongoDB, auditing tracks database activities like authentication, authorization, and data access, while encryption protects data at rest and in transit. Auditing helps with security, compliance, and troubleshooting, while encryption ensures data confidentiality.

Multi-cloud & Distributed architecture differences :

Multi-cloud refers to using cloud services from multiple providers, while distributed architecture involves distributing workloads across multiple nodes within a single or multiple cloud environments. Multi-cloud focuses on vendor flexibility and risk mitigation, whereas distributed architecture prioritizes scalability and resilience.

Key Differences:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Multi-cloud** | **Distributed Architecture** |
| Focus | Vendor flexibility, risk mitigation | Scalability, resilience, cost efficiency |
| Resources | Multiple cloud providers | Multiple nodes within a single or multiple clouds |
| Management | Potentially more complex due to different providers | Can be complex due to resource coordination |
| Application | Running applications across multiple clouds | Distributing workloads within a single application |
| Example | Using AWS and Azure for different parts of a website | A database distributed across multiple servers |

Cloud :

MongoDB offers cloud services primarily through MongoDB Atlas, a fully managed, multi-cloud developer data platform. It allows users to deploy, scale, and manage MongoDB databases in the cloud, eliminating the need for on-premises infrastructure and manual management tasks. MongoDB also provides services like Atlas Search and Atlas Data Federation for enhancing data capabilities in the cloud.

Here's a more detailed look:

MongoDB Atlas:

This is a cloud database service that simplifies deploying and managing MongoDB databases. It supports multi-cloud deployments, meaning you can run your databases on AWS, Azure, or Google Cloud.

Fully Managed:

MongoDB Atlas handles tasks like provisioning, backups, scaling, and performance optimization, allowing you to focus on application development.

Multi-Cloud:

You can deploy your Atlas databases across multiple cloud providers, providing geographic flexibility and resilience.

Additional Services:

MongoDB also offers services like Atlas Search (full-text search) and Atlas Data Federation (federated database instances) to enhance your data capabilities in the cloud.

Benefits of using MongoDB in the cloud:

Scalability: Easily scale your database up or down based on your needs.

Flexibility: Adapt your database infrastructure to changing application requirements.

Efficiency: Automate database management tasks to reduce operational overhead.

Cost-effectiveness: Pay-as-you-go pricing model allows you to optimize costs.

Jenkins & Apache -Tomact installation:

First console output checked :

