MODULE 11

Praneesh Sharma

21052264

**Amazon ElastiCache:** A web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud. The service improves the performance of web applications by letting you retrieve information from fast, managed, in-memory caches, instead of relying on slower disk-based databases.

**Cache:** In computing, a cache is a high-speed data storage layer that stores a subset of data, typically transient in nature, so that future requests for that data are served up faster than is possible by accessing the data’s primary storage location.

**Data caching:** Storing data in a cache lets you efficiently reuse previously retrieved or computed data. The data in a cache is generally stored in fast-access hardware such as random access memory (RAM) and can also be used with a software component.

**Elastic Load Balancing:** Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon Elastic Compute Cloud (Amazon EC2) instances, containers, IP addresses, and AWS Lambda functions. If traffic to a website suddenly spikes, that traffic can be routed to other EC2 instances (or other types of instances such as Lambda instances) that have been established in advance for this purpose. This load balancing avoids a single server being overloaded because of increased traffic routed to it.

**Random Access Memory (RAM):** Volatile, temporary memory storage. This is the data that is held temporarily while a machine is in use; however, once the machine is powered off or the task is completed, this data goes away. Virtual memory is stored in the read-only memory (ROM) as a supplement to RAM when there is not enough temporary memory available.

A data caching service such as ElastiCache can help to ensure that the data can be accessed and processed extremely quickly. It works by storing the data in extremely fast but temporary memory that is faster than disk-based storage. The trade-off is that the fast memory has less storage space and does not store the data permanently. Many companies use ElastiCache to build real-time apps, speed up ecommerce, and cache their websites.

Heavy traffic can shut down apps and websites if the server cannot handle the load. This is why AWS has ELB, which can detect when there are too many requests and automatically divert traffic into a new server to maintain speed and stability. There are three types of ELB in AWS.

****Application Load Balancer:**** Application Load Balancer is best suited for load balancing of Hypertext Transfer Protocol (HTTP) and Secure HTTP (HTTPS) traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request. Application Load Balancer balancing is done based on the content of the uniform resource locator (URL).

****Network Load Balancer:**** Network Load Balancer is best suited for load balancing of Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Transport Layer Security (TLS) traffic where extreme performance is required. Operating at the connection level (Layer 4), Network Load Balancer routes traffic to targets within Amazon VPC and is capable of handling millions of requests per second while maintaining ultra-low latencies. Network Load Balancer is also optimized to handle sudden and volatile traffic patterns. Because of the increased speed that can be achieved at the connection layer, the Network Load Balancer type of load balancing is more desirable when trying to avoid higher volumes of network traffic.

****Classic Load Balancer:**** Classic Load Balancer provides basic load balancing across multiple EC2 instances and operates at the request and connection levels. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

**LAB 11: Using Load Balancers**

1. Services > EC2
2. Launch an instance named ‘Web Server 1’ with key pair as ‘vockey’ and a custom network security group and a custom HTML script for the user data
3. Instances > Web Server 1 > Actions > Images and templates > Launch more like this
4. Use the same configurations as Web server 1 to make a new instance ‘Web Server 2’
5. Load Balancing > Load Balancers > Create Load Balancer > Application Load Balancer > Create
6. Name the balancer as ‘myloadbalancer’ with a custom target group ‘myalbTG’