High Availability:

high availability is a technique to avoid the interrupts like (s/w & h/w failures), process failures, minimize downtime and provide services even in interrupts.

some of the key concepts to achieve high availability are:

1)Redundancy: redundancy means if any component fail duplicate severs, db , or components will take the action to process the request.

2)Clustering: grouping set of processes to work together to build scalable and reliable s/y's

3)Failover: failover is a mechanism it is used to automatically divert the failure to backup s/y or components. If one severs fail load balancer may divert the traffic to backup s/y's or components to continue service.

4)Distributed S/y’s: high availability can be achieved by designing distributed s/y’s, different components of application run on separate servers. If one server fail other will continue to handle the req.

5)Load balancing: distributing incoming reqs to multiple servers to reduce the load on single server. Load balancing can be implemented at various levels like DNS- load balancing(is a type of load balancing that uses the DNS to distribute the traffic across several servers) or load balancing devices(Azure application gateway , AWS Elastic load balancing).

6)Monitoring and Health checks: monitoring and health checks helps to detect the issues early. s/y's are designed to react automatically to identified problems, failover.

Auto-Scaling:

Automatically adjusting the capacity of application, auto scaling achieves that application can handle the load at peak hours.by using auto-scaling, application will be able to respond fast to req, reduce the latency.

Automatically adjusting resources based on usage it helps to reduce the cost.

Auto-scaling commonly used in cloud computing platforms such as AWS, AZURE, Google cloud.

It is an important feature for building scalable and resilient application.

Tools:

* Amazon Ec2 auto scaling
* Google Cloud Auto Ml
* Azure auto scale

Resilience:

A resilient technology s/y can quickly adapt changes, recover from failure continue to provide services even when unexpected events occur.

Redundancy: if a s/y fail backups are used to process the req.

Backup: If data loss or s/y fail, organizations can quickly restore their data through back up.

Load balancing: distributing incoming reqs to multiple servers to reduce the load on single server. Load balancing can be implemented at various levels like DNS- load balancing (is a type of load balancing that uses the DNS to distribute the traffic across several servers) or load balancing devices (Azure application gateway, AWS Elastic load balance.

Tools:

* Apache JMeter
* Pioneer
* Pumba

Observability

Observability is understanding, monitoring, analyse the internal behaviour of the s/y.

Observability provides information what is happening within a s/y, it helps the developer to identify the issues resolves issues.

Observability mainly focus on four factors:

* Logging
* Monitoring
* Tracing
* metrics

Benefits of Observability:

Discover and address unknown issues: Observability discovers conditions you might never know or think to look

Catch and resolves issues early: Developer teams can identify and fix issues in new code before they impact the customer experience

**Scale observability automatically: the ability of a system to dynamically adapt and expand its observability capabilities in response to changes in workload, complexity, or other environmental factors.**

**Tools:**

* **Splunk**
* **Grafana**
* **Dynatrace**
* **App dynamics**

**Monitoring**

**Processes needed to track, review, regulate the progress, performance of the application.**

Some other examples of performance monitoring are:

* Monitoring CPU of server
* Tracking and monitoring error rate
* Monitoring n/w traffic

Tools:

* Graylags
* Splunk
* New Relic
* Datadog

Logging

* Logging means recording events, activities, and information about the execution of a application.
* Logs are typically generated during the runtime of the application and serve as a valuable tool for developers, system administrators, and support teams to understand and troubleshoot the application's behaviour.
* Logging is an essential aspect of observability and is used for various purposes, including debugging, performance monitoring, error tracking, and auditing.

Tools:

* Log4j
* Logback
* Java.util. logging

Alerts

* Alerts are notifications or messages by predefined conditions or events within the application.
* The purpose of alerts is to notify developers about issues, anomalies, or situations that require attention.
* Alerts play a crucial role in maintaining the reliability, performance, and security of applications.

Types of Alerts:

* Error Alerts
* Performance Alerts
* Availability Alerts
* Security Alerts

Tools:

* Alertmanager
* PagerDuty
* Grafana
* Datadog

RTO(Recovery Time Objective)

The  recovery time objective (RTO)is the amount of time or real time during or after a disaster that can elapse without a business restoring its services and processes to acceptable levels before it will experience intolerable consequences associated with the disruption.

The OSI Reference Model

The OSI model has seven layers.

The Physical Layer: The physical layer is concerned with transmitting raw bits over a communication channel.

* The design issues have to do with making sure that when one side sends a 1 bit, it is received by the other side as a 1 bit, not as a 0 bit.

The Data Link Layer: The main task of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer.

* It accomplishes this task by having the sender break up the input data into data frames and transmits the frames sequentially

The Network Layer: The network layer controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination.

* Routes can be based on static tables that are ''wired into'' the network and rarely changed.

transport layer:

The transport layer also determines what type of service to provide to the session layer, and, ultimately, to the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent.

The Session Layer:

The session layer allows users on different machines to establish sessions between them.

Sessions offer various services, including dialog control token management and synchronization.

The Presentation Layer:

The presentation layer is concerned with the syntax and semantics of the information transmitted.

The presentation layer manages these abstract data structures

and allows higher-level data structures

The Application Layer:

The application layer contains a variety of protocols that are commonly needed by users.

One widely-used application protocol is HTTP (Hypertext Transfer Protocol), which is the basis forth World Wide Web.

Recovery point objective (RPO)

Recovery point objective (RPO) is defined as the maximum amount of data – as measured by time – that can be lost after a recovery from a disaster, failure, or comparable event before data loss will exceed what is acceptable to an organization.