**3 TIER ARCHITECTURE:**

The three-tier architecture is a design pattern commonly used in web application development that separates an application into three logical tiers: presentation, application, and data. In the context of cloud computing, the three-tier architecture can be implemented using virtual machines or containers in a cloud environment.

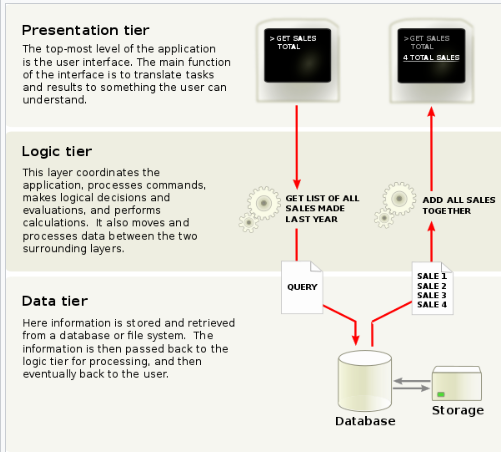
The presentation tier is the front-end layer of the application that interacts with the user. It includes the user interface components such as web pages, forms, and graphical elements. This tier is responsible for receiving user requests and returning responses.

The application tier is the middle layer of the application that processes user requests and generates responses. It includes business logic, processing algorithms, and other application components. This tier is responsible for handling application-specific functions and processing data from the presentation tier.

The data tier is the back-end layer of the application that stores and manages data used by the application. It includes databases, file systems, and other data storage systems. This tier is responsible for persisting application data and ensuring its availability.

In a cloud environment, the three tiers can be implemented as separate virtual machines or containers running on a cloud platform such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP). Each tier can be scaled independently, and load balancing can be used to distribute traffic across multiple instances for improved performance and availability.

Overall, the three-tier architecture in cloud computing provides a scalable, flexible, and reliable approach to deploying web applications.



**Reference link**: <https://en.wikipedia.org/wiki/Multitier_architecture>

**benefits of 3-tier architecture:**

Scalability: The 3-tier architecture is designed to be scalable. Each tier can be scaled independently to handle increasing amounts of traffic, allowing for more efficient use of resources.

Maintainability: Separating an application into layers makes it easier to maintain and update. Changes made to one tier do not affect the other tiers, reducing the risk of unintended consequences.

Security: The 3-tier architecture enhances security by separating sensitive data from the presentation layer, making it more difficult for attackers to access and compromise the data.

Flexibility: Each tier can be developed and updated independently, giving developers more flexibility to work on specific areas of an application without disrupting other areas.

Reusability: The 3-tier architecture promotes code reusability by separating functionality into distinct layers. This can save development time and reduce errors, as well as make it easier to add new features to an application.

Performance: The 3-tier architecture can improve application performance by distributing the workload across multiple servers or machines. This can reduce response times and improve the user experience.

Overall, the 3-tier architecture provides a flexible, scalable, and secure framework for building robust and maintainable software applications.

**problems arised by 3 tier architecture in cloud:**

While the 3-tier architecture can provide many benefits in terms of scalability, maintainability, security, flexibility, reusability, and performance, there are also some challenges that can arise when implementing this architecture in the cloud. Some of these challenges include:

Network Latency: In a cloud environment, the communication between different tiers of an application can be slowed down by network latency. This can be particularly problematic if the application requires real-time interactions between tiers, such as in the case of an online game or a financial trading system.

Dependency Management: In a cloud environment, the different tiers of an application may be hosted on different machines or even in different geographic regions. This can make it difficult to manage dependencies between tiers and ensure that all components of the application are working together as expected.

Security Risks: In a cloud environment, there may be additional security risks associated with the distribution of data and services across multiple machines or geographic regions. This can make it more difficult to control access to sensitive data and prevent unauthorized access or data breaches.

Cost: While the cloud can provide many benefits in terms of scalability and performance, it can also be expensive to operate. Applications built using the 3-tier architecture may require additional resources, such as load balancers or additional servers, to operate effectively in a cloud environment.

**GENERAL STEPS:**

Choose a cloud platform: First, you need to select a cloud platform that suits your needs. Some popular options are Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

Create a virtual machine: Once you have chosen a cloud platform, you need to create a virtual machine (VM) for each tier of the application. The VMs can be created using the cloud platform's management console or command-line interface.

Install and configure software: After creating the VMs, you need to install and configure the software for each tier. For example, you would install a web server like Apache or Nginx on the front-end VM, a database like MySQL or MongoDB on the back-end VM, and an application server like Tomcat or JBoss on the middle-tier VM.

Connect the tiers: Next, you need to connect the tiers so that they can communicate with each other. You can use tools like load balancers, firewalls, and virtual networks to create secure connections between the tiers.

Deploy the application: Once everything is set up, you can deploy the web application to the middle-tier VM. You can use tools like Git, Jenkins, or Ansible to automate the deployment process.

Test and monitor: Finally, you should test the application to ensure it is working as expected. You should also monitor the application using tools like CloudWatch, Azure Monitor, or Stackdriver to detect and fix any issues that arise.