

1. A paragraph on what PaaS, SaaS and IaaS are and the differences between them.

Why do we need cloud computing service?

Acquiring computing resources to run applications or store data the traditional way requires time and capital. Organizations must purchase equipment through procurement processes that can take months. They must invest in physical spaces, typically specialized rooms with power and cooling. And after deploying the systems, they need IT professionals to manage and maintain them. All this is challenging to scale when demand spikes or business grows. Also, you run the risk of running out of capacity or overbuilding and paying for infrastructure that you never use.

Cloud computing makes hardware and software resources available on the Internet and are managed as third party services. These services typically provide access to everything from simple backup applications to advanced server virtualization.

And the biggest advantage of cloud computing is that it helps in increasing capacity and capabilities without requiring any additional infrastructure to be added – providing great savings.

IaaS:

Infrastructure as a service (IaaS) is a type of cloud computing service that offers essential compute, storage, and networking resources on demand, on a pay-as-you-go basis.

IaaS cloud servers are typically offered to businesses over the internet, whether through a dashboard or an API, ensuring users have complete control over their computing infrastructure.

- Advantage:

1. Cost savings: IaaS helps reduce your upfront capital expenditures. Resources are used on demand, meaning you only have to pay for compute, storage, and networking resources that you consume. IaaS costs are fairly predictable and can be easily contained and budgeted for.
2. Increased efficiency: IaaS resources are regularly available to businesses when they need them. As a result, organizations can reduce provisioning delays when expanding infrastructure and, alternatively, avoid wasting resources by overbuilding capacity.
3. High scalability: IaaS has the capability to automatically scale resources up and down rapidly. You can accommodate sudden spikes in demand almost instantly and scale back when the resources are no longer required.
4. Reliability: IaaS platforms have no single point of failure. Cloud infrastructure offers built-in redundancy and fault tolerance, with workloads spread across multiple servers and facilities. Even if one component of the hardware resources fails, the service will usually remain available.

- Downside:

1. Legacy systems: Before a full migration to the cloud is completed, any legacy technology or applications should be reviewed for compatibility. There are many older systems that are not designed for cloud-based services and may need to be upgraded or replaced.
 2. Security: With the move from on-premise to the cloud, there could be new security threats, whether sources from the host or other virtual machines (VMs). It is critical that organizations review and research up-to-date security threats and their remediation strategies.
 3. Internal Training: With a new system comes a lack of familiarity with its intricacies. Businesses should prepare additional training and resources to ensure their users know what they are doing.
- Use cases:
1. Run testing and development
 2. Improve disaster recovery preparation
 3. Perform big data analysis
 4. Handle spikes in traffic
 5. Provision resources quickly

PaaS:

Like IaaS, for PaaS, the service provider delivers and manages the backend infrastructure. However, PaaS models provide all the software features and tools needed for application development. You still have to write the code and manage your apps and data but do not have to worry about managing or maintaining the software development platform.

PaaS provides that platform for software developers to create, allowing them to concentrate on the software itself instead of any external issues. But you need your own logic to take advantage of PaaS to process the data properly. Use it if you need a lot of codes for development. PaaS solutions are specific to application and software development and typically include:

- Cloud infrastructure: Data centers, storage, network equipment, and servers
- Middleware software: Operating systems, frameworks, development kits (SDK), libraries
- User interface: Graphical user interface (GUI), Command line interface, an API interface

- Advantage:
1. Faster time to market: No heavy lifts required. Developers have instant access to a complete application development platform that they don't have to build or manage, freeing up time to develop and deploy.
 2. Low maintenance: In-house application stacks come with headaches, especially when it comes to upgrades. With PaaS, the provider is responsible for keeping everything up-to-date—and none of the maintenance pain is yours

3. Cost-effective pricing: PaaS resources are on-demand, so you only pay for what you actually use. A PaaS also provides access to advanced development tools and capabilities that might be too expensive to purchase outright.
 4. Easy scalability: No more worrying about capacity, and lets you scale down for low-traffic periods / scale up immediately to meet unexpected surges in demand.
- **Downside:**
 1. Integrations: With PaaS, you may encounter challenges when integrating new applications. This is tied into issues related to legacy systems since there are often aspects of these systems that aren't built for the cloud.
 2. Data security: Using third-party servers for your data could potentially lead to additional security risks. Security options may also be limited since you will have to find a solution that can integrate with the third-party systems.
 3. Runtime: PaaS solutions may not be fully optimized for the language and frameworks your business uses and finding a specifically tailored solution may be difficult.
 - **Use case**: PaaS is often the most cost-effective and time-effective way for a developer to create a unique application. PaaS allows the developer to focus on the creative side of app development as opposed to the menial tasks such as managing software updates or security patches. Instead, their time and brainpower can go into creating, testing and deploying the app. A good non-ecommerce example of PaaS is AWS Elastic Beanstalk. A good ecommerce example is Adobe Commerce (Magento).

SaaS:

Software as service (SaaS) is the most commonly used service within the cloud market. They are typically ready-to-use and run from a users' web browser. SaaS providers deliver the entire application and infrastructure with full functional service, and they are responsible for everything like managing the data, servers and storage. Users don't need to worry about anything like storage, IT review and streamlining business processes, additional downloads or application installations, etc. Google Docs is a great example of SAAS.

- **Advantage:**
 1. Easy Access: you don't need to install and run software applications on your computer. Everything is available over the internet when you log in to your account online. You can usually access the software from any device, anytime — as long as there is an internet connection.
 2. Convenience to use: All your staff will have personalized logins suitable to their access level. You no longer need to engage an IT specialist to download the software onto multiple computers throughout your office or worry about keeping up-to-date software on every computer. It's all taken care of in the Cloud.

3. Payment Structure: Most SaaS providers operate a subscription model with a fixed, inclusive monthly account fee. You know precisely how much the software will cost and can budget accordingly without worrying about hidden surprises
- **Downside:**
 1. **Customization**: SaaS services typically allow minimal customization for features, capabilities and integrations. This can force companies to invest significant resources into managing or adding customization capabilities.
 2. **Lack of control**: With a SaaS solution, businesses often have to hand over ultimate control to the third-party service providers — giving them the keys to functionality, performance and even data. You will want to ensure that you trust the provider used or have the ability to review your information remotely.
 3. **Interoperability**: Integrations with existing applications and services can be a concern since many SaaS apps are not designed for open integrations. Finding a service with integration capabilities can be difficult, and attempting to create your own can be worse.
 4. **Data Security**
 - **Use Case**: Used when you already have an end product that is ready for consumers, and when you want minimum coding and development. Consumers don't care about platforms and hardwares. An example will be gmail (email service), so we don't have to do any coding or set up an email server. And we don't care about where data(emails and attachments) are stored. Google will take care of that for consumers.

IaaS vs PaaS vs SaaS:

- IaaS is there to provide you with maximum flexibility when it comes to hosting custom-built apps, as well as providing a general data center for data storage.
 - IaaS examples: Compute Engine, Cloud Storage.
- PaaS is most often built on top of an IaaS platform to reduce the need for system administration. It allows you to focus on app development instead of infrastructure management.
 - PaaS example: Cloud Run, App Engine.
- SaaS offers ready-to-use, out-of-the-box solutions that meet a particular business need (such as a website or email). Most modern SaaS platforms are built on IaaS or PaaS platforms.
 - SaaS example: Google Workspace.

2. A paragraph on the differences between ETL and ELT. Also, list the pros and cons of each in a chart. And specify when you'll use which.

<https://www.xplenty.com/blog/etl-vs-elt/>

ETL:

Extract, Transform, Load process is used when you need to bring data from different data sources into one centralized database. One simple example of this is managing sales data in shopping malls. If users want the historical data as well as current data in the shopping mall, the first step is the user needs to follow the ETL process. Then that data will be used for reporting purposes. Also, banking data, insurance claims.

- Extract: EXTRACT data from its original source
 - Source like CRM system/Data storage platforms/cloud
- Transform: TRANSFORM data by cleaning, processing, combining to ensure quality
 - Cleansing, Standardization, Deduplication, Sorting
- LOAD data into the target database
 - load the newly transformed data into a new destination (data lake or data warehouse.) Data can be loaded all at once (full load) or at scheduled intervals (incremental load)
- Advantage:
 1. ETL allows for speedier, more efficient, more stable data analysis.
 2. ETL provides a more secure way to perform transformations (like transforming emails to just the domain or removing the last part of an IP address) because it transforms the data before putting it into the data warehouse.
 3. More well-developed ETL tools and platforms are available to assist with data extraction, transformation, and loading needs. Also, data engineers skilled and experienced at setting up ETL pipelines are easy to find.
- Use case:
 1. Used when you task requires speedy analysis
 2. Used when companies relate to compliance and regulated by GDPR, HIPAA, or CCPA so that you need to remove, mask, or encrypt specific data fields to protect the privacy of their clients.

ELT:

Extract, Load, and Transform process. Data gets leveraged via a data warehouse in order to do basic transformations. That means there's no need for data staging. ELT uses cloud-based data

warehousing solutions for all different types of data - including structured, unstructured, semi-structured, and even raw data types

- Advantage:

1. High Speed: When it comes to data availability, ELT is the faster option. ELT allows for all of the data to go into the system immediately, and from there, users can determine the exact data they need to both transform and analyze.
2. Low-Maintenance: With ELT, users generally won't have to have a "high-touch" maintenance plan. Since ELT is cloud-based, it utilizes automated solutions instead of relying on the user to initiate manual updates.
3. Quicker Loading: Because the transformation step doesn't occur until after the data has entered the warehouse, it cuts down on the time it takes to load the data into its final location. There's no need to wait for the data to be cleansed or otherwise modified, and it only needs to go into the target system once.

- Use Case:

1. Used when you want to process those huge amounts of data more quickly
2. Used when An organization with the resources to handle the processing power needed. With ETL, the majority of the processing takes place while the data is still in pipeline before it gets to your warehouse. ELT does its work once the data has already arrived in the data lake. Depending on what needs to be done to the data to suit your purposes, smaller companies may not have the financial flexibility to develop or explore the extensive technology needed to get the full benefits of a data lake.
3. Used when a company needs all its data in one place as soon as possible. When the transformations take place at the end of the process, ELT prioritizes the speed of transfer over almost everything else, which means that all data - good, bad, and otherwise - ends up in the data lake for later transformation