Aspect	Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
Definition	Provides virtualized infrastructure components like servers, storage, and networks.	Provides a managed platform with tools and frameworks for developing and deploying applications.	Delivers ready-to-use software applications over the internet.
Purpose	For organizations to manage and control their computing resources flexibly.	For developers to focus on building applications without managing infrastructure.	For end users to use pre-built software solutions without managing underlying components.
Key Features	- Virtual machines and storage - Network configurations - Scalability - Pay-as-you-go	- Development tools - Managed hosting - Middleware and runtime environments - Integrated CI/CD pipelines	- Accessible via web - No installation required - Subscription-based - Automatic updates
User Responsibility	Manage OS, middleware, runtime, applications, and data.	Manage applications and data while the platform handles infrastructure and runtime.	Use the software with no responsibility for maintenance or updates.
Control	High control over infrastructure.	Limited control over infrastructure, high focus on applications.	No control over infrastructure or platform.
Scalability	Highly scalable infrastructure.	Scalable for application hosting and development needs.	Scales to meet the number of users and application needs.

Setup Complexity	High, as it requires configuration and management of virtual machines and networks.	Medium, as infrastructure is abstracted, but application management is required.	Low, as no setup or configuration is needed.
Examples	- **AWS EC2**: Virtual servers. - **Microsoft Azure VM**: Virtual machines. - **Google Compute Engine**: Virtualized compute resources.	- **AWS Elastic Beanstalk**: Deploy web applications. - **Google App Engine**: Build apps on a managed platform. - **Azure App Service**: Host and deploy web apps.	- **Google Workspace**: Cloud-based productivity tools. - **Salesforce**: CRM software. - **Zoom**: Video conferencing.
Applications	- Hosting virtual machines - Disaster recovery - Big data analysis - Test and development environments	- Building custom applications - Developing APIs - Collaborative app development	- Email and communication tools - CRM platforms - ERP systems - Online storage (e.g., Dropbox)
Cost Model	Pay-as-you-go, based on the use of computing, storage, and network resources.	Pay-as-you-go, based on the platform's usage time and resources consumed.	Subscription-based or license-based.
Target Users	IT teams, system administrators, DevOps engineers.	Developers, application architects, and startups needing rapid deployment.	Business end users, non-technical teams, and consumers.

Advantages	- Full control over the environment. - Cost-efficient for businesses with IT expertise. - Highly customizable.	- Speeds up development cycles. - Managed environment. - Easy integration with CI/CD pipelines.	- No maintenance or updates. - Accessible from anywhere. - High ease of use.
Disadvantages	- Requires technical expertise. - High complexity in configuration and maintenance.	- Limited infrastructure control. - Vendor lock-in risks. - Dependency on the platform's limitations.	- Limited customization. - Can become costly with large-scale usage. - Privacy concerns with third-party software.
Protocols/Technologies	Virtualization (VMWare, Xen), Kubernetes, Docker.	Application frameworks (Spring, Django), CI/CD tools (Jenkins), APIs.	HTTPS, OAuth, REST APIs, multi-tenancy frameworks.
When to Use?	- When you need complete control over hardware and software stack. - Custom workloads or specific requirements. - Hosting complex applications.	- When developing and deploying apps without managing infrastructure. - Rapid prototyping and testing.	- When you need ready-to-use software. - For collaboration, CRM, or email needs. - Reduce operational costs.