Click to prove you're human

Examples of platform as a service

Platform as a Service (PaaS) is a cloud computing model that provides developers with a ready-to-use platform for building, deploying, and managing applications. It eliminates the need to manage underlying infrastructure, such as servers, storage, and networking, by offering a suite of tools, services, and APIs that streamline the development process. Rather than code everything from the ground up, PaaS providers often have pre-built blocks that developers can just plug and play to build better apps quickly. PaaS is ideal for developers looking to streamline the development process, as it automates tasks such as software updates, security patches, and resource scaling. Kinsta has been a managed WordPress hosting provider well known for its high-performance hosting platform and top-notch customer support. The company has recently introduced Application Hosting, providing customers with a comprehensive Platform-as-a-Service (PaaS) solution. With this new offering, Kinsta takes care of complex tasks such as infrastructure setup, container orchestration, custom domain integration, and server management, allowing customers to focus on developing and growing their applications. By entrusting these time-consuming responsibilities to Kinsta’s expert team, customers can rely on a reliable, one-stop-shop solution for their hosting needs. Cloudways is perhaps unique on this list because it is extremely deeply rooted in the web hosting industry. Although it is just like many other PaaS platforms and offers users high configurability for rapid deployment, many have used it to instead build custom virtual servers for hosting. Part of the reason is its provision for managed hosting, which combines the power of Cloud PaaS without the overly technical portion of environment management. Pricing models are as transparent as many others as well. Microsoft Azure is no stranger to cloud dominance, and Azure App Services is its answer to the need for a streamlined PaaS solution. Designed to help businesses deploy web applications and APIs without hassle, App Services simplifies hosting, scaling, and security management. It’s a favorite among enterprises that rely on .NET, Java, and Node.js applications. Plus, with built-in integration for DevOps workflows, CI/CD pipelines, and enterprise security features, it’s easy to see why Azure App Services is a go-to choice for scalable cloud apps. By abstracting much of the infrastructure management, Azure App Services lets businesses focus on building applications that deliver value—rather than dealing with server headaches. Heroku now belongs to Salesforce and is an example of PaaS based on the managed container concept. As with many PaaS environments, it is highly self-contained and integrates data services as well as a complete ecosystem of its own. Because of its app-centricity, Heroku has gained a reputation as less of an enterprise solution. Instead, it has gained a following among the hobbyist and production development crowd. It also helps that Heroku is pretty user-friendly, allowing it to offer a more streamlined experience. For those keen to build on Heroku, we’ve seen apps built on this platform that has managed to go for respectable prices. For example, The Regular, built to sell food and drinks, is being listed on Flippa for around $25,000. When you think about Amazon Web Services (AWS), cloud computing comes to mind, and AWS Elastic Beanstalk is part of that expansive ecosystem. Unlike raw infrastructure services, Elastic Beanstalk is a managed PaaS solution designed to streamline application deployment. Developers can deploy applications quickly without worrying about underlying infrastructure. It supports popular programming languages like Java, Python, Node.js, and more, automatically handling scaling, monitoring, and load balancing. The best part? You still retain control over underlying AWS resources if you need deeper customization. Elastic Beanstalk is particularly useful for teams that want to focus on development without getting bogged down by complex server configurations. Just push your code, and AWS takes care of the rest. Google offers their App Engine as part of the Google Cloud ecosystem. It’s intended to be a highly scalable serverless PaaS used for rapid deployment. Google, being the giant it is, can provision highly capable servers capable of coping with almost any volume of query. There have, however, been some issues raised by developers about the service. These include a slight lack of support in some language environments, a shortage of development tools, an inability to plug-and-play some applications, plus a lock-in to Google as the vendor. Lauding itself as the “smallest PaaS implementation you’ve ever seen”, Dokku – a PaaS example that isn’t quite as capable as the big players like AWS. What it lacks for in depth however, it makes up for in cost – Dokku is open source and completely free. Based on container technology from Docker, this minute PaaS essentially lets you deploy on any infrastructure. The serious advantage of this is that there is a much lower chance of vendor lock-in so you’ll be able to take your business model in any direction you want. Fly.io takes a unique approach to cloud deployment, focusing on running applications close to users for ultra-low latency. Unlike traditional PaaS providers that centralize hosting in a few regions, Fly.io lets developers deploy globally with minimal configuration. Built primarily for containerized applications, Fly.io makes it easy to run full-stack apps with minimal DevOps overhead. It automatically routes traffic to the nearest instance, optimizing speed and performance for users across different locations. Lightning is what Salesforce considers the next generation of their platform. It’s separate from Salesforce classic (which is SaaS) and will be the target of all future Salesforce developments in future. Lightning offers a much-improved user interface and has improvements that will boost the experience of both business users as well as on the IT team side. An important part of the rapid development feature is the inclusion of reusable building blocks and a new delivery system. IBM Cloud Foundry brings open-source flexibility to the PaaS space, catering to developers who want a fast and scalable environment for building cloud applications. It supports multiple programming languages, including Java, Ruby, and Go, making it a versatile choice. What makes IBM Cloud Foundry stand out is its focus on portability and automation. Developers can push applications with minimal setup, while Cloud Foundry handles provisioning, load balancing, and scaling behind the scenes. If you’re looking for a platform that combines open-source agility with enterprise-grade reliability, IBM Cloud Foundry is a solid contender. With IBM’s backing, it offers strong security and support—ideal for businesses that need a dependable cloud application environment. OpenShift is in a way similar to Cloudways and offers users an easier way to build and deploy applications on. It also has extensive API support so you aren’t limited only to what the platform has to offer. Coming from Red Hat, OpenShift has also been known to be incredibly secure. There are multiple safeguards built into the environment which will step in should users try to perform unexpected actions (such as attempting to run containers with incorrect permissions). Render is a rising star in the PaaS space, offering a modern alternative to traditional cloud providers. It provides developers with fully managed hosting for web applications, static sites, databases, and background workers – without the operational headaches. What makes Render stand out is its simplicity. With an intuitive UI and straightforward pricing, developers can deploy applications with minimal effort. Whether you’re running a Next.js site, a Python API, or a PostgreSQL database, Render handles scaling and security in the background. For startups and small teams looking for a hassle-free cloud platform, Render strikes the right balance between ease of use and performance. Platform.sh is a PaaS designed with developers in mind, offering a powerful cloud hosting solution that automates infrastructure management. Unlike traditional cloud hosting, Platform.sh focuses on Git-based workflows, allowing teams to deploy applications effortlessly across multiple environments. With built-in support for popular frameworks like Symfony, Laravel, and Django, Platform.sh simplifies the development cycle. It enables instant cloning of production environments for testing, making it a strong choice for teams that need fast iteration without the hassle of manual server configurations. If your team values efficiency, scalability, and automated deployments, Platform.sh offers a powerful platform to keep projects moving forward without distractions. Wasabi may not be the size of Google, Amazon, or Oracle but it is one of the larger independent PaaS providers in the market. Their very impressive pricing has made them popular for a much wider variety of use cases such as Cloud storage. It has a simple interface and is easy to use and set up as well. This makes it a good choice for both personal use and small to medium-sized businesses. The convenience in these situations makes up for those having more limited access to strong technical support teams. Not sure which web host is the best fit for your website? Get tailored hosting advice from our team today. Click the button below to begin. Whether you have questions about performance, pricing, or specific features, we’re here to help you make the best decision. Click Here Cloud computing services are commonly categorized into three main models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Each model serves a different purpose, offering varying levels of control, scalability, and ease of use. PaaS, as we discussed in this article, simplifies development by offering a managed platform where developers can build, deploy, and run websites or applications without dealing with the underlying infrastructure. IaaS provides the fundamental building blocks of cloud computing, including virtual servers, networking, and storage. Businesses using IaaS have full control over their infrastructure, allowing them to configure and manage their own operating systems and applications. This model is ideal for companies that need flexible, scalable cloud hosting solutions, such as running websites, virtual machines, or enterprise applications. Examples of IaaS providers include Digital Ocean, and Kamatera. SaaS is the most user-friendly cloud model, providing fully managed software applications over the internet. Users do not need to install, update, or maintain anything—everything is handled by the provider. Common SaaS applications include email services, collaboration tools, and customer relationship management (CRM) software. Examples SaaS offerings include Hostinger Website Builder and Spanel. PaaS is ideal for businesses that develop custom applications but may not be necessary for companies that simply need a website, email hosting, or an online store. Business hosting is often the better choice for these cases. Business hosting offers managed environments optimized for CMS platforms, email, and online business operations. PaaS, on the other hand, provides developer tools for building custom applications but may require more technical expertise. Yes, but migration can be complex. If you’re currently using PaaS and realizing you don’t need its advanced infrastructure, moving to a business hosting provider can be a cost-effective and easier-to-manage solution. Common use cases for PaaS include developing and hosting web applications, creating and deploying APIs, developing mobile applications, and implementing data analytics solutions. PaaS is especially useful for startups and small businesses that may not have the resources to set up and manage their own infrastructure, as well as for organizations looking to quickly prototype and test new applications. PaaS platforms are game-changers for developers looking to deploy and scale applications without managing infrastructure. Whether you’re using Kinsta, Cloudways, or Azure App Services – these solutions simplify the development process. However, if your primary goal is running a website – not developing custom applications – PaaS may not be necessary. Many small businesses, startups, and non-technical users don’t need cloud-based developer platforms. Instead, they require a reliable, easy-to-manage business hosting solution that handles: Website hosting (WordPress, Joomla, Magento, etc.) Business email hosting and domain management Scalability for growing traffic without cloud complexity Strong security features like SSL, backups, and DDoS protection This is where business hosting comes in. Unlike PaaS, which focuses on application development, business web hosting is designed for companies that need fast, secure, and hassle-free website management—without the technical overhead. Get your Hong Kong business account open in a few days100% online applicationNo account opening fee, no initial depositAccount managerPhysical and virtual cards Platform as a Service (PaaS) is a cloud computing model that provides a comprehensive platform for developing, running, and managing applications without the complexity of building and maintaining the underlying infrastructure. Incorporating a social media marketing communication plan can enhance the promotion and visibility of applications developed on a PaaS. Additionally, integrating an Ecommerce chatbot can improve customer interaction and support for businesses using PaaS. Utilizing social media mass communication strategies helps in reaching a broader audience and driving engagement. Finally, generating a service receipt ensures clear documentation of the services provided, facilitating transparency and accountability between the service provider and the client. Platform as a Service (PaaS) is a cloud-based solution that provides developers with the tools and environment needed to build, deploy, and manage applications. Instead of managing hardware and software infrastructure, developers use PaaS to access pre-configured development tools, runtime environments, and middleware. To ensure transparency and accountability, a service performance report can be generated to evaluate the efficiency and effectiveness of the platform services. A service list details all available services within the PaaS offering, helping developers choose the right tools for their projects. When specific services are requested, a service order outlines the terms and scope of these services. For maintenance tasks, a cleaning service report may be utilized to document the upkeep and management of the platform environment. Google App Engine Microsoft Azure App Service Amazon Web Services (AWS) Elastic Beanstalk Heroku IBM Cloud Foundry Oracle Cloud Platform Red Hat OpenShift Salesforce App Cloud SAP Cloud Platform Mendix Zoho Creator Engine Yard OutSystems Pivotal Cloud Foundry Apache Stratos Description: Hosted on the public cloud and available for anyone to use. Examples: Google App Engine, Microsoft Azure, Heroku. Description: Deployed within a company’s private network or data center. Examples: Red Hat OpenShift, VMware Tanzu. Description: Combines both public and private PaaS environments, allowing data and applications to be shared between them. Examples: IBM Cloud Pak, Salesforce Lightning Platform. Description: Provides a platform for integrating communication features like voice, video, and messaging into applications. Examples: Twilio, Nexmo, Plivo. Description: Focuses on providing the tools and environment to develop mobile applications. Examples: Firebase, IBM MobileFirst, Kinvey. Description: Provides tools to integrate applications and data across different environments. Examples: MuleSoft Anypoint Platform, Dell Boomi, Informatica. Development Tools: IDEs, version control, build tools. Middleware: Application servers, message brokers, API management. Database Management: Managed database services. Application Hosting: Environments for deploying and running applications. Operating Systems: Underlying OS support. Networking: Load balancing, DNS, virtual networks. Security: Firewalls, IAM, encryption. Analytics and Monitoring: Performance tracking and diagnostics. Collaboration Tools: Team collaboration and project management. APIs and Integrations: Connecting with other services and applications. Scalability and Load Balancing: Auto-scaling and load balancing features. Backup and Recovery: Data backup and disaster recovery services. Support and Maintenance: Technical support and system maintenance. Provisioning the Platform: Developers sign up and get access to a pre-configured platform. Development Environment: Provides online tools for coding, testing, and debugging. Middleware Services: Connects the application to necessary backend services. Application Deployment: Automates deployment, scaling, and load balancing. Database Management: Offers easy-to-use, scalable database services. Scaling and Load Balancing: Automatically adjusts resources based on demand. Security and Compliance: Integrates security features and ensures compliance. Monitoring and Analytics: Provides tools for tracking performance and usage. CI/CD Pipelines: Automates code integration and deployment processes. Backup and Recovery: Ensures data protection with automated backup and recovery. Support and Maintenance: Provider manages infrastructure updates and support. CTO: Strategic oversight and alignment with business goals. IT Director: Integration and management of PaaS within the IT infrastructure. DevOps Manager: Automation and operational integration of PaaS. Cloud Architect: Design and architecture of cloud solutions. Development Team Lead: Guidance and support for developers using PaaS. Systems Administrator: Technical management of the PaaS environment. Product Manager: Alignment of PaaS with product goals. Security Officer: Ensuring compliance and security standards. Project Manager: Managing PaaS-related projects and deliverables. AspectModern PaaSTraditional PaaSDeployment ModelCloud-native, often supporting hybrid and multi-cloud deployments.Primarily single-cloud or on-premises deployment.ScalabilityAutomatically scales resources based on demand.Limited auto-scaling capabilities, often requires manual scaling.DevOps IntegrationSeamless integration with DevOps tools and practices (CI/CD, IaC).Basic or limited integration with DevOps tools.Microservices SupportBuilt to support microservices architectures and containerization.Primarily designed for monolithic applications.Container OrchestrationOften includes support for Kubernetes and Docker.Little to no built-in support for container orchestration.APIs and IntegrationRich API ecosystem and easier integration with third-party services.Limited APIs and more difficult third-party integrations.User ExperienceFocus on developer experience with streamlined, user-friendly interfaces.More complex interfaces, less focus on user experience.Cost EfficiencyPay-as-you-go pricing, optimized for cost efficiency.Fixed pricing models, potentially higher costs.Security FeaturesAdvanced security features like encryption, IAM, and compliance tools.Basic security features, often requiring additional configuration.Analytics and MonitoringAdvanced monitoring, logging, and analytics tools included.Basic monitoring and logging, often requiring third-party tools.Support for Emerging TechIntegrates emerging technologies like AI/ML, IoT.Limited support for emerging technologies.Flexibility and CustomizationHighly customizable environments and extensive configuration options.Less flexibility, more standardized environments.Deployment SpeedFaster deployment cycles with continuous delivery support.Slower deployment cycles, often manual processes. Growth and Expansion Description: The PaaS market is expected to continue growing rapidly, driven by increasing adoption of cloud services and digital transformation initiatives. Projection: Analysts predict significant market expansion, with revenues reaching hundreds of billions of dollars in the next few years. Hybrid and Multi-Cloud Solutions Description: Organizations are increasingly adopting hybrid and multi-cloud strategies to avoid vendor lock-in and leverage the best services from multiple providers. Impact: PaaS providers will focus on offering seamless integration and management across different cloud environments. Edge Computing Integration Description: The rise of edge computing will push PaaS providers to offer solutions that support distributed computing closer to data sources. Impact: Enhanced performance, lower latency, and better data handling for applications requiring real-time processing. AI and Machine Learning Description: PaaS platforms will increasingly integrate AI and machine learning capabilities to offer intelligent automation and advanced analytics. Impact: Improved decision-making, predictive analytics, and personalized user experiences. Serverless Architecture Description: Serverless computing will become a more prominent feature of PaaS offerings, allowing developers to focus on code without managing infrastructure. Impact: Increased developer productivity and cost-efficiency. Security and Compliance Description: Enhanced focus on security, compliance, and data privacy due to rising cyber threats and regulatory requirements. Impact: PaaS providers will offer advanced security features and compliance certifications. Developer Experience Description: Emphasis on improving the developer experience with better tools, documentation, and community support. Impact: Faster development cycles, higher satisfaction, and increased adoption rates. Subscription-Based Pricing Description: Continued dominance of subscription-based models with tiered pricing based on usage, features, and support levels. Impact: Predictable revenue streams and scalable pricing for businesses of all sizes. Pay-as-You-Go Description: Flexible, consumption-based pricing models that charge based on actual usage of resources. Impact: Cost-efficiency and scalability for users, particularly startups and SMBs. Freemium and Free Trials Description: Offering basic features for free with the option to upgrade to premium plans. Impact: Lower entry barriers, allowing users to test and adopt the platform before committing financially. Revenue Sharing and Marketplace Models Description: Platforms offering marketplaces for third-party apps and integrations, with revenue-sharing arrangements. Impact: Expanded ecosystem, additional revenue streams, and enhanced platform capabilities. Bundled Services Description: Bundling PaaS with other cloud services (IaaS, SaaS) to provide comprehensive solutions. Impact: Increased value proposition and simplified procurement for customers. Custom Solutions and Enterprise Plans Description: Tailored solutions and pricing for large enterprises with specific needs and high-volume usage. Impact: Greater customization, dedicated support, and strategic partnerships. AspectInfrastructure as a Service (IaaS)Platform as a Service (PaaS)Software as a Service (SaaS)DefinitionProvides virtualized computing resources over the internet.Provides a platform allowing customers to develop, run, and manage applications.Delivers software applications over the internet on a subscription basis.Components Managed by ProviderNetworking, storage, servers, virtualization.Networking, storage, servers, virtualization, OS, middleware, runtime.Networking, storage, servers, virtualization, OS, middleware, runtime, data, applications.Components Managed by UserApplications, data, runtime, middleware, OS.Applications, data.None (fully managed by the provider).Use CasesData storage, backups, disaster recovery, virtual machines.Application development, deployment, and testing.End-user applications like email, CRM, collaboration tools.ExamplesAmazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP).Google App Engine, Microsoft Azure App Services, Heroku.Google Workspace (formerly G Suite), Microsoft 365, Salesforce.ScalabilityHigh, users can scale resources up or down as needed.High, users can scale applications easily within the platform.High, users can add or remove users and features as needed.Cost StructurePay-as-you-go based on resource usage.Subscription or pay-as-you-go based on usage.Subscription-based, often with tiered pricing.Control LevelHigh control over the infrastructure and environment.Moderate control over the application environment.Low control, primarily end-user configuration.Setup and ManagementRequires IT expertise to manage and configure infrastructure.Easier setup than IaaS, but requires knowledge of application development.Simplest setup and management, user-friendly interfaces.MaintenanceUser is responsible for managing and maintaining applications and data.Provider manages the infrastructure and platform, user manages applications.Provider handles all maintenance and updates. Advantages: Rapid development and deployment. Cost savings on infrastructure and maintenance. Scalability to handle variable workloads. Focus on application development. Integrated development tools and services. Automated updates and maintenance. Global accessibility for remote teams. Disadvantages: Risk of vendor lock-in. Limited customization options. Security and compliance concerns. Potential for downtime and outages. Performance constraints due to shared resources. Ongoing cost management challenges. Complexity in integrating with existing systems. PaaS enables multiple developers to work on the same project simultaneously by providing a shared environment and tools. PaaS supports multiple programming languages, including Java, Python, Ruby, PHP, and .NET. PaaS platforms automatically scale resources up or down based on application demand, ensuring optimal performance. PaaS providers implement security measures like data encryption, firewalls, identity management, and regular security updates. Yes, PaaS can integrate with various cloud services, including databases, storage, and third-party APIs, enhancing functionality. PaaS supports DevOps by providing continuous integration, continuous deployment, and automated testing tools for streamlined workflows. Examples include Google App Engine, Microsoft Azure App Services, and AWS Elastic Beanstalk. Yes, you can deploy existing applications on PaaS platforms with minimal modifications, depending on compatibility. PaaS platforms offer managed databases and storage solutions, simplifying data management and backup processes. PaaS typically uses a pay-as-you-go pricing model, charging based on resource usage, such as compute power, storage, and data transfer. Add Tone Friendly Formal Casual Instructive Professional Empathetic Humorous Serious Optimistic Neutral 10 Examples of Public speaking 20 Examples of Gas lighting Platform as a service (PaaS) is defined as a cloud computing platform where a third party offers the necessary software and hardware resources. 14 PaaS Examples In Real LifeConclusion These offerings enable clients to develop, run, and manage business applications without maintaining the infrastructure required for such software development processes. In this article, we will be giving some PaaS examples in real life as well as how they carry out their work and their relevance. The cloud services offered by today’s leading PaaS providers are equipped to operate across languages, libraries, containers, and equivalent tools. These include computing, storage, databases, developer and management tools, and security. Some of PaaS examples in real life include: 1. Microsoft Azure Microsoft Azure is a cloud computing platform developed by Microsoft. It offers management, access and development of applications and services to individuals, companies, and governments through its global infrastructure. Microsoft Azure is perhaps one of the most approved PaaS examples that you will come across. This is most likely because it is used widely across the world. It is so expansive that it includes all three basic cloud models. Furthermore, it can support every step in the app development process, from the first piece of code to the final deployment of the code. It supports many programming languages, tools, and frameworks, including Microsoft-specific and third-party software and systems. Microsoft Azure allows an individual the ability to code in ASP.NET, PHP, or Node.js. It helps firms develop the simplest capability to easily migrate all their apps and infrastructure. It includes mobile services as well as advanced applications in business analytics. 2. Heroku Heroku is a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. It is not as much an enterprise platform as a service solution as it is a self-contained system that is centered around apps. It also has space for data integration. It is also mostly made up of ecosystems accessible entirely within the platform. It is known for its intuitive and user-friendly UI. Its ease of use has made it a favorite among production developers and those developing apps just for fun. The Developer Experience Module included in Heroku allows one to deploy directly from popular repositories such as GitHub and Git and contains a dashboard that allows you to add apps and utilities to your solution easily. It is regularly updated and patched and is compatible with a wide host of languages. 3. AWS Lambda AWS Lambda is a part of the AWS Cloud and is another PaaS example in real life. It is completely integrated with all the different AWS services and has a server-less architecture. If you want to build custom backend services that can be triggered on demand through the use of custom API endpoints, then this is a wonderful option for you. It is widely integrated with nearly every third-party library and language. The platform’s administration is completely automated, allowing users to focus only on the services they intend to provide. The platform is highly fault-tolerant and developed to protect the user’s code against machine language errors, allowing it to run smoothly and without a hitch. AWS Lambda can automatically scale to the level of incoming support requests. 4. Google App Engine Google App Engine is among the many PaaS examples that are part of a whole suite of services. It is a part of the Google Cloud and offers highly scalable apps and tools that can be used as a part of your service. It is compatible with many languages, even though some key options seem to have been missed. It has a standard environment, which makes it easy to build and deploy an application that runs reliably even under heavy loads and with large amounts of data. It is serverless and available continuously without any downtime. 5. Dokku Dokku is the smallest PaaS platform that has ever been created. It is an extensible, open source platform as a service that runs on a single server of your choice. Due to its small size, the features it offers to developers are limited. However, its high user base may be attributed to its being completely free and open source. It helps an individual build and manage the lifecycle of applications, from building to scaling. It is built on a container-based Docker platform. It offers wide integration, even though scalability is expectedly limited. It offers multiple methods of deployment and guides you through every step of the deployment process. It also helps you with configuration and network management. 6. Apprenda Cloud Platform Apprenda Cloud Platform is one of those enterprise PaaS examples that borrows from open-source solutions to improve their robustness. The unique selling point of Apprenda Cloud Platform is its ability to transition services from traditional applications running on dot-net to a PaaS environment. It allows you to rapidly deploy and run applications built on Java as well as dot net. The developer and the IT operations in the platform are completely automated, and control is placed in the hands of the programmers. Programmers also get the capability to define the security policy of their applications. It allows you to run multiple clouds, whether private or public. 7. Pivotal Cloud Foundry Pivotal Cloud Foundry is an example of PaaS managed by the Cloud Foundry Foundation. It is a distribution of the open-source Cloud Foundry platform. It is a powerful framework for the fast development and deployment of applications. The workflows of Cloud Foundry are highly streamlined, and the framework itself is very scalable. It allows for portability and automatic scaling of applications. In addition, it also contains tools for application health management and infrastructure security. 8. Salesforce Lightning Salesforce has always been highly preferred among platforms as a service, but Salesforce Lightning is considered the next generation of the PaaS platform. It offers a host of new features and great updates to old ones. It offers a much-improved user interface that is dashboard based and easy to use. It contains interactive tabs as well as an activity timeline that make the task of rapid application development and deployment much easier for both the frontend users and the backend IT side. Salesforce lightning’s homepage is a great example of simplicity, even in the most complex of applications. The most loved feature of Salesforce Lightning has been the virtual assistant, which isn’t just very helpful but also seems capable of getting you exactly what you want. 9. IBM Cloud Foundry IBM Cloud Foundry is an open-source PaaS example that is still extremely powerful. While it can’t be called nearly as capable as its other counterparts developed by IT majors, it does offer some really useful features and, over the years, has been scaling up to get into the big leagues. IBM Cloud Foundry supports various languages, from Java to PHP and Python to Ruby. In addition, their build packs are regularly updated on the IBM Cloud Foundry Community. It is also highly fault-tolerant and allows your application to run smoothly and without hotches caused by code errors. IBM Cloud Foundry is an open-source version of IBM’s PaaS, which is powerful as well as agile. Cloud Foundry allows coordinated linkage of coding parameters (related to building and deployment) to associated cloud services. As a consequence, applications are deployed quickly and consistently. 10. Red Hat OpenShift Red Hat OpenShift is supposed to provide users with a significantly easier platform to create and deploy applications. It is compatible with a wide range of APIs, ensuring that the user is never restricted just to what is included in the platform. There are various features that Red Hat OpenShift has to offer to its users. It is based on Kubernetes, and all its elements are made available on the cloud. It contains a simple user interface, and perhaps its greatest feature is its security options to stave off intrusion and prevent unauthorised access internally. Red Hat’s OpenShift offers users a seamless way to build and deploy applications. It also provides extensive API support, thereby extending its capability beyond the platform itself. It is also known to be an incredibly secure platform. 11. Oracle Cloud Platform The Oracle Cloud Platform is yet another among the PaaS examples in cloud computing used primarily at the enterprise level. Oracle is an industry that has high expertise in all aspects of the cloud. Oracle Cloud Platform allows you to run Oracle as well as non-Oracle applications smoothly on its servers. It is highly secure and offers great flexibility with APIs. It is also great at providing analytical tools on the cloud, which can improve the applications you have built. You can also choose the method you want to use for greater flexibility in the deployment process. PaaS service is one of the leading products among those offered by the company. The platform was initially designed to work with Oracle SaaS applications, but it also works with other apps. 12. Zoho Creator Zoho Creator is perhaps the simplest to use among the PaaS examples you will find here. It is not a very complex platform, nor does it have any highly sophisticated tools or utilities, but its ease of use makes it a considerably popular choice among beginners and learners. It is as easy as dragging and dropping tools in a dashboard and is centered around making application development a visual process rather than one based on code. It also contains quite a few useful automation options. 13. Wasabi Wasabi is a platform-as-a service example meant for individual use or for small to medium-sized businesses. Wasabi is easy to set up and has a clean user interface, making application development a task that takes much less time than usual. It is among the platform service providers that are not yet enterprise-scale yet contain many highly scalable utilities and features. 14. Cloudways Cloudways is among the platform-as-a service providers that use web hosting as a strategy. While it contains useful tools for rapid application development and deployment, it has also been widely used to create virtual web servers. This is probably due to the feature of managed web hosting, which makes the task much simpler. It is not open source but is pretty transparent about its features, pricing, and code compatibility. Conclusion PaaS technology was first introduced by Salesforce.com, Heroku and Amazon Web Services (AWS) back in 2007. After this , Google launched App Engine in 2008, which was a free trial version. Overnight, these initiatives transformed the online cloud computing space into a complete virtual industry. It was offered as a complete solution for an enterprise-customized Research and Development (R&D) middleware platform. Now PaaS technology has evolved and not only offers application software but also covers both application and database services. PaaS cloud technology has changed the development, testing and deployment processes of service-oriented architecture (SOA). It has also provided ease in the application development process for the whole SOA architecture. PaaS platform usage will grow in the coming years as more citizen developers use them to create applications.