Infosys cobalt

WHITE PAPER

ACCELERATE APPLICATION MODERNIZATION WITH CLOUD-NATIVE DEVELOPMENT

Abstract

Cloud has become the de-facto environment for modern applications. These days new-age applications are not only born but also managed across their entire lifecycle on the cloud. This has propelled the emergence of cloud-native technologies that are primarily open source. In a way, cloud has spurred the rapid adoption of open source technologies. This evolution in technology and application management has armed businesses with more power and opportunities that can be used to lead the competition. This paper explains the approaches and techniques for resolving key challenges in becoming cloud-native. It also focuses on how Infosys can help its customers accelerate cloud-native application development.

Infosys®
Navigate your next

Introduction

Enterprises across the globe are experiencing a massive disruption in their business models, led by the forces of nature as well as radical technology innovations. The physical footprints of customers are fading away, giving rise to a digital customer identity. In a world where customers are identified by their clicks, which may come from any part of the world, a single moment of business non-availability can have adverse and cascading consequences. Industries must be prepared to cater to the elastic nature of demand.

Cloud enabled industries to ensure greater availability and on-demand elasticity without owning a single piece of infrastructure. This made hosting applications on the cloud increasingly popular. Soon, a new genre of applications known as cloud-native applications came into existence. The design and architecture of these applications significantly differ from traditional cloud-hosted applications. Their unique architecture principles allow them to exploit cloud capabilities to the fullest.

Understanding the cloud adoption path

When it comes to cloud adoption, each enterprise has its own imperatives driven by business and IT needs. Enterprises typically go through three phases of cloud adoption. However, at any point, enterprises can be in all these three phases at the same time based on application criticality, agility, etc.

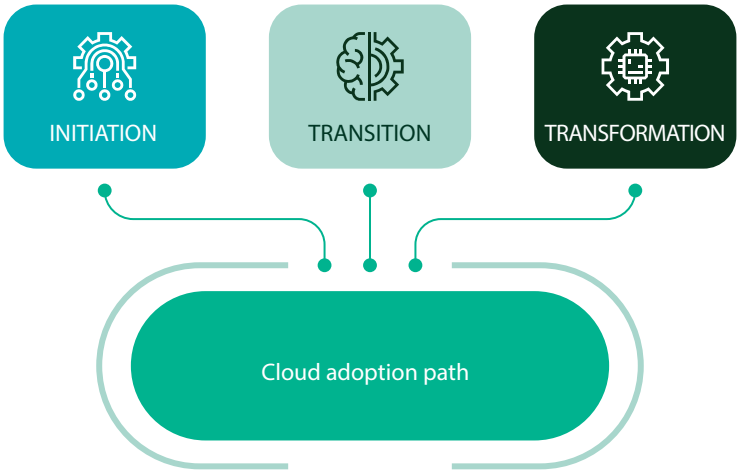


Fig 1: Key phases of cloud adoption

- **Initiation** – This is the first and fundamental building block in the evolution of the IT landscape to cloud. Cost reduction, resiliency and rapid cloud adoption are the key objectives. The primary adoption scenario is lifting and shifting workloads to cloud due to data center consolidation, requiring minimal changes to applications. In this phase, applications typically do not take advantage of native cloud features and it may even cost more to run them on cloud.
- **Transition** – In this phase, only parts of the application are modified. The key objective is to address existing application pain points or improve specific functional areas to quickly gain business benefit using only those cloud features that are necessary to achieve this. Some key adoption patterns that apply to this phase are legacy data offload to cloud and legacy API-fication.
- **Transformation** – This phase focuses on modernizing core technical capabilities of the application landscape while migrating to cloud, to meet dynamic business demands. It aims at improving business agility and customer experience by adopting cloud through application transformation, standardization and automation techniques. The primary focus of this phase is on building and architecting applications on cloud that are lightweight and distributed in nature. Here, applications offer higher performance and are optimized to operate at lower cost.

The shift to cloud-native

There are many modernization themes that can be applied across the three adoption phases. While many organizations are still doing lift and shift, most cloud adoption programs aim to be cloud-native through application-centric transformation approaches.

Cloud-native is an approach to building and running applications that exploit the advantages of the cloud computing delivery model without being tied down to any specific cloud platform.

Cloud-native technologies empower organizations to build and run scalable applications in modern dynamic environments such as public, private and hybrid clouds. Containers, microservices, APIs, DevSecOps, and continuous delivery practices are the key pillars of cloud-native development. These applications are designed as loosely coupled and stateless microservices, packaged in lightweight containers. These microservices are exposed through APIs for interaction and collaboration. Each microservice goes through an independent lifecycle managed by agile and DevSecOps processes. They are highly automated, starting from infrastructure provisioning to application deployment. Multiple continuous delivery pipelines can work in tandem to deploy and manage cloud-native applications.

These technologies and architectural patterns help build loosely coupled systems that are resilient, manageable and observable, and deliver benefits like:

- **Improved time to market** through an accelerated application development lifecycle, enabled by container and modern DevSecOps processes
- **Cost advantages** by using open source technologies and elastic containers for managing applications at scale
- **Enhanced customer experience** through APIs and highly available mobile-first applications
- **Avoiding vendor lock-in** by moving out of proprietary software and embracing open source

Fig 2: Key elements of cloud-native technologies

Trends in cloud-native application development

Cloud-native development is changing how enterprise applications are developed and deployed. Given below are some of the key trends in cloud-native development.

- **New ways of application development and deployment** – Monolithic applications played a crucial role in traditional application development. Such applications have resource intensive workloads and are prone to development and deployment errors. Microservices patterns and adoption of serverless, event-driven, reactive applications are gaining momentum. The miniature size of these applications encourages adoption of new developer tools that are shifting from locally deployed thick IDEs to cloud-hosted lightweight workbenches. This provides a standard and error-free experience to all developers.
- **Improved orchestration and management techniques** – Cloud-native development supports automation of previously manual tasks like load balancing, scaling and self-healing, thereby enabling applications to react and respond in real-time. With the advent of technologies like service mesh, microservices orchestration is further enhanced with additional capabilities of service discovery and load balancing. Developers or administrators no longer need to write code to enable inter-container communication or schedule deployments.
- **PaaS and native cloud integration** – It is important to choose the right platform for enterprise applications. Cloud and enterprise container platforms focus on a scale-out approach and use the 'build once, deploy anywhere' principle. The ability of a container platform to be hosted on public or multi-cloud environments and integrate with native cloud services like DevOps, Database, IAM, etc., has opened new possibilities of utilizing the best of both cloud and container worlds.
- **Cloud-native infrastructure** – Enterprises are adopting simple, repeatable and automated provisioning of infrastructure, platform and applications. Infrastructure-as-code embedded with DevSecOps has become an integral part in setting up environments within no time. Managed Kubernetes-based offerings provide a standard infrastructure-less user experience and abstract the complexity of infrastructure provisioning. This has also enabled faster time to market when it comes to developing cloud-native applications.
- **Observability and analytics at scale** – To keep pace with change, enterprises need to respond to forces within their ecosystem. While observability is important in cloud-native applications and has gained maturity over the years, the recent trend is to adopt analytics as well. This enables organizations or application teams to detect and respond faster, thereby inching towards becoming a live enterprise. With the use of advanced machine learning on observability data, it is possible to understand failure patterns and trend analysis much faster compared to traditional application environments.

"By 2023, more than 70% of global organizations will be running more than two containerized applications in production, up from less than 20% in 2019."

- Gartner [1]

- **Kubernetes at the edge** – Kubernetes had traditionally been a platform for on-premise environments. The buzz around IoT and edge-based computing is pushing Kubernetes towards the edge, making it run on low resources and in cloud agnostic environments. With Kubernetes, organizations can run containers at the edge in a way that maximizes resources, makes testing easier and allows DevOps teams to move faster and more effectively as these organizations consume and analyze more data in the field.
- **AI/ML workloads on Kubernetes** – Machine learning services have been successful with some of the major public cloud providers. With the growth of Kubernetes adoption on cloud and edge, organizations are looking to deploy their machine learning business models on Kubernetes and reap the benefits of its scale-in feature combined with cloud's scale-out feature.

Cloud-native maturity model

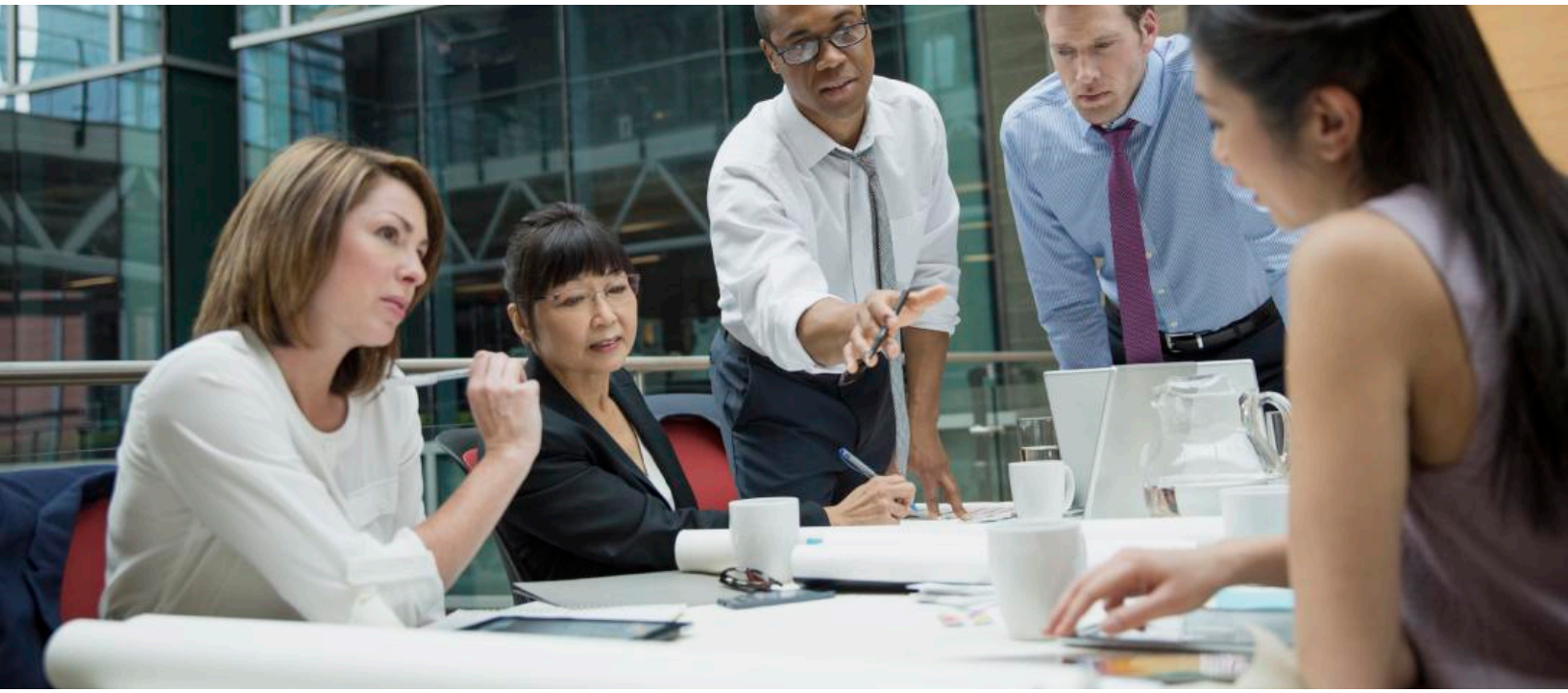
The digital maturity of an enterprise can be determined based on where it stands in its cloud-native adoption journey. The three main categories are:

- **Watchers** – These organizations start small by migrating existing applications to cloud or container platforms. They are beginning to adopt APIs, DevSecOps and automation through best practices. However, cloud-native technologies are often deployed in silos across peripheral processes. They are more experimental in nature to identify the fitment of various cloud-native capabilities. Some of the patterns seen in these organizations are:
  a. Lift-tinker-shift a monolithic application onto a container platform
  b. Evolution of DevOps culture and practice
  c. API-fication on a refactored monolith
- **Explorers** – Most organizations are in this stage where they want to re-engineer applications to cloud-native standards. They tend to have a mature DevSecOps pipeline with cloud services like DBaaS, iPaaS, etc. Here, applications are decomposed into microservices and effective communication patterns are established. Some of the patterns seen in these organizations are:
  a. Adopting native cloud managed services
  b. Enhancing DevSecOps automation with infrastructure configuration and deployment alongside applications that accelerate the development lifecycle
  c. Functional decomposition of monoliths into microservices
- **Visionaries** – These organizations are quite advanced in their cloud-native journeys due to aggressive adoption of new technologies like serverless, event-driven architecture along with strong governance for security and compliance. They aim to achieve hyper-productivity through self-service and extreme automation portals. Some of the patterns that could apply to these organizations are:
  a. Adoption of emerging technologies on Kubernetes – serverless, event driven, service mesh, and AI/ML
  b. Hyper-productivity through a self-service portal enabling extreme automation
  c. Continuous delivery using advanced deployment (A/B, Canary, Blue-Green)
  d. Moving to cloud-native infrastructure

Cloud-native adoption challenges

While there are guidelines and best practices on how to adopt cloud-native application development, the actual process of creating a production-ready cloud-native platform can be difficult. Some of the main challenges faced by organizations are:

- **Choosing the right technology** – With numerous technology options available in the market, organizations find it extremely challenging to select the best fit as per their requirement. Some common concerns are choosing the right architecture and technology stack that align with the enterprise IT strategy, and ensuring a cohesive development platform with several vendor products.
- **Lack of subject matter experts** – Most enterprises search for specialists in more than one area. However, with the ever-changing technology landscape and components, gaining expertise even in one area is difficult. There is a dearth of experts who fully understand the intricacies of even one capability area mentioned in the previous section.
- **Too many things to stitch together** – Cloud-native application development is built on loosely coupled design principles and distributed across multiple runtimes, clusters and zones. It becomes difficult to bring all these together and provide a single unified view. Lack of proper automation can lead to significant effort wasted in unnecessary hand-offs between multiple teams.



Role of an enterprise container platform

An enterprise container platform provides developers with an open environment to quickly build cloud-native applications and address some of the challenges above. A container platform consists of three elements that accelerate cloud-native development. These are:

The Cloud Native Computing Foundation offers over 650 technologies to guide organizations on cloud-native adoption[2]

- **Container at the core** – Containers help organizations build once, deploy anywhere. There are multiple runtimes available within a container, making it easier to run polyglots when compared to traditional environments. By default, container platforms also support governance and lifecycle management.
- **Microservices and cloud-native application support** – Container platforms support microservices and provide mechanisms for integrating and coordinating loosely coupled services. They provide enhanced application management capabilities and higher application resilience by isolating developers from server and operating system dependencies. These platforms also provide alerts and dashboards for applications as well as support for cloud-based integration.
- **Automation capabilities** – Container platforms support high automation capabilities. At the application level, these provide automated, centralized and consolidated logging and monitoring as well as automated orchestration of containers for high availability and scalability. These platforms also provide DevSecOps support and align with the concepts of infrastructure and pipeline as code.

Infosys approach to cloud-native application development

While enterprise container platforms address some of the cloud-native adoption challenges, these are still insufficient to achieve the hyper-productivity that organizations want. In order to achieve this, Infosys proposes a four-step approach:

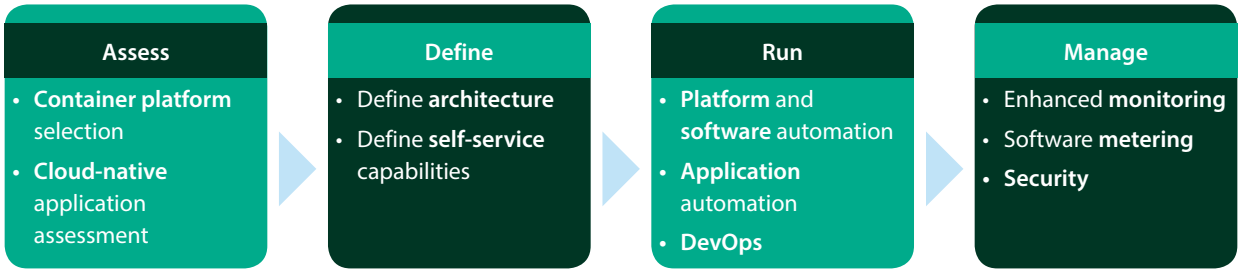


Fig 3: Building blocks for cloud-native development

Assess

- **Container platform selection** – The first step towards cloud-native adoption is choosing the right container platform. This is an elaborate exercise as multiple inputs go into deciding the container platform. The as-is application technology landscape, technology and business needs, cloud deployment models are the key inputs to this exercise. In addition, the out of box capabilities and features of a container platform, influenced by community and enterprise support, help in deciding the right platform.
- **Cloud-native application assessment** – Before migrating or architecting applications to cloud-native principles, teams need to undergo multiple assessment exercises. These assessments can help in finding the container suitability and decide on the disposition while migrating to container platforms. They can also be about discovering the right-sized platform and environment before hosting cloud-native applications. Cloud-native platforms should facilitate these essential capabilities in order to support the subsequent phases of development.

Define

- **Define architecture** – Most cloud-native platforms use a product-first approach, which depends heavily on any specific vendor's product stack. An architecture-first approach is vendor-agnostic, enabling unbiased decisions and choices. Since multiple open source products promise to solve similar problems, it becomes necessary to have a product evaluation framework to guide organizations in making the right architecture decisions. Once the technology stack is selected, organizations can then define and create templates of prescribed architecture patterns based on selected products. The templates should cover multiple architecture patterns pertaining to software, application, DevOps, security, etc. Architects and developers can then select from the list of available and recommended architecture patterns.
- **Define self-service capabilities** – It is important to define a persona-driven workbench for architects, developers, infrastructure operations, gatekeepers, and approvers to carry out daily activities before the start of the execution phase. A cloud-native platform should provide highly automated self-service capabilities to all personas to facilitate onboarding new projects, developing microservices and creating pipelines along with supporting services like logging, monitoring and tracing. This is needed to improve productivity and efficiency by minimizing dependencies on manual, centralized processes. The platform should also be able to support common technical and domain services for developers to link with the application.

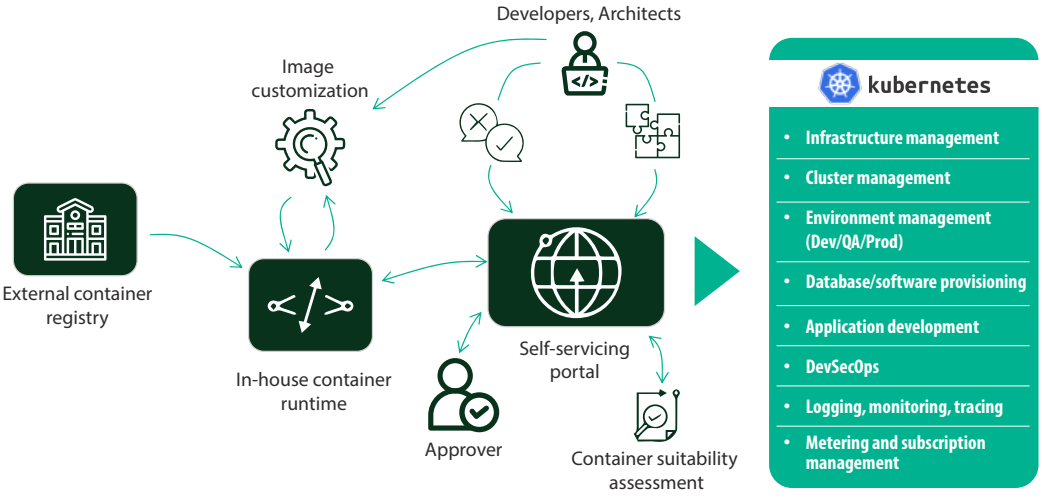


Fig 4: How self-service portals improve productivity

Run

Platform and software automation – While most container platforms provide production-ready images, these require significant effort to ensure seamless software lifecycle management. Software products have their own configuration parameters and understanding these along with the underlying technology takes time. Thus, an ideal cloud-native platform should support a template-driven method for software deployment and management, together with automated deployment tools.

Application automation – A cloud-native platform should enable developers to produce any scaffolding code in an interactive way while creating user experience layers, microservices and backing services. Microservices developers should be able to add/modify DevSecOps pipelines and bind the software with the services without needing to write or modify code. All the developer productivity functions should be integrated in a workflow to achieve hyper productivity.

DevSecOps – Even though the organization has defined tools and technology stacks for DevSecOps, it can take a long time to onboard new applications in their existing DevSecOps environment. A cloud-native platform should aim to reduce the traditional time taken and enable architects/developers to onboard applications to the existing DevSecOps environments and create pipelines within minutes.

Manage

Enhanced monitoring – While all cloud-native platforms provide integrated monitoring dashboards, developers still have a hard time finding out relevant information. Developers need customizable dashboards that display application-related information. Moreover, these platforms should possess AI-driven capabilities that continuously improve the monitoring experience by sensing and responding to events.

Software metering – A disadvantage in modern cloud-native platforms is the inability to meter the use of enterprise open source software against its subscription. Product vendors are unaware of how these software subscriptions can be effectively metered against their correct usage in a volatile container world. A cloud-native platform should have a codified way to track open source software subscription, consolidate cloud and software utilization, and provide a single view of consumption and cost.

Security – Application security and vulnerability are top concerns, particularly in cloud-native applications. Distributed processes in these applications pose new security challenges. The attack surface area becomes much larger due to the number of technologies and services in addition to the increased communication between services. Additionally, in the open source world, it can be challenging to keep track of new vulnerabilities, bugs, patches, and releases across communities. While enterprise container platforms support all layers of security, architects and administrators need configure and integrate multiple security products and technologies. Certain security capabilities needs to be configured and managed outside application development phase viz. credential management, image scanning, governance policies, runtime security and other security checks built into the DevSecOps automation lifecycle. Integration with vulnerability databases and open source project communities for near real-time updates can further enable application security and stability.

Infosys Cloud Native Development Platform

Infosys has encoded its expertise, experience and best practices from working with global customers into a powerful, yet lightweight and adaptable platform that simplifies and accelerates the entire cloud-native journey for enterprises.

Using an architecture-first and developer-centric approach, the Infosys Cloud Native Development Platform abstracts and automates much of the complexity that goes into building cloud-native applications on any Kubernetes-based platform. Built-in open source product evaluation cuts down the time needed to define the right architectural components. With automated provisioning and configuration of full stack application environments, enterprises can get their projects started in hours. Developers can focus only on writing business logic, while the platform generates all the scaffolding code for microservices and UI, creates DevSecOps pipelines and configures end-to-end security and observability.

Rich dashboards enable developers and administrators to stay on top of application performance as well as new product vulnerabilities, updates and discussions in open source communities.

Since technologies are constantly evolving, the platform is built to adapt to changes. New technologies and processes can easily be plugged in. At the same time, it does not create any kind of proprietary lock-in. Everything created is open and can continue to be used even without the platform.

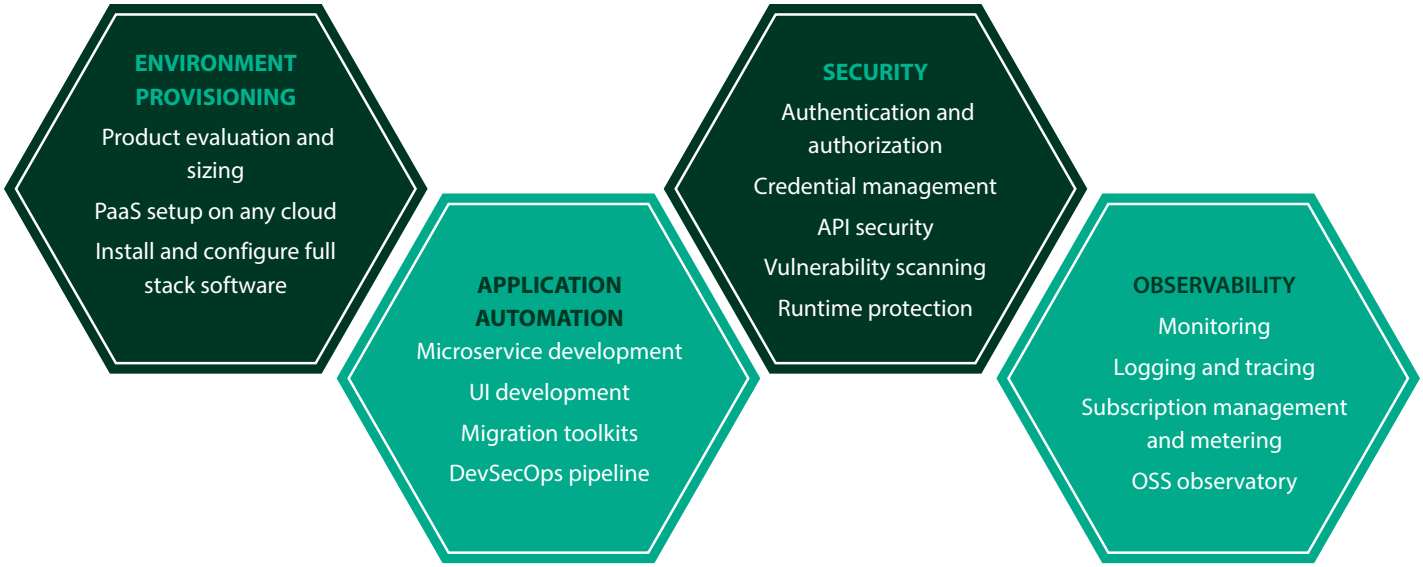


Fig 5: Features of Infosys Cloud Native Development Platform

Adopting the Infosys Cloud Native Development Platform helps organizations achieve productivity gains as well as enhance developer experience and operational effectiveness. The platform helps in driving modernization initiatives that enable enterprises to:

Containerize and modernize existing applications – Accelerate brownfield application development on containers

Build new applications – Boost agility and efficiency while developing greenfield cloud-native applications

Automate operations – Fastrack IT operations by bringing together data from multiple disconnected sources to produce appropriate information, eliminate risk and reduce cost.

Conclusion

As we navigate the digital economy, the key to being more agile lies in harnessing the power of cloud and breaking free from monolithic legacy applications. Cloud-native applications can help unlock the full value of the cloud by moving from infrastructure-centric to application-centric models. However, with many cloud-native developments trends on the rise, it is difficult to gain expertise and keep up with new technology choices. To become visionaries in the cloud-native world and deliver business applications in a faster, simpler and more consistent manner, enterprises need to build a strong foundation centered around architecture-first, developer-centric, automation and agile principles. The Infosys Cloud Native Development Platform implements these principles through a new approach to architecting, developing and deploying applications in a manner that simplifies and accelerates the cloud-native journey on any Kubernetes platform. It provides one-click creation of full stack environments in minutes and up to 40% faster microservice development. Enterprises can further accelerate the transformation through one-click deployment of Infosys industry solutions. The platform codifies Infosys' experience and best practices using extreme automation techniques, thereby fueling the modernization journey and keeping organizations future ready.

About the authors

Somnath Banerjee, *Principal Technology Architect at Infosys*

Somnath Banerjee is a Principal Technology Architect with the Infosys Modernization Practice. He leads the cloud-native development solutions for the practice and has been with Infosys for around 16 years. He has strong experience in developing and delivering enterprise applications in multiple complex client engagements across several verticals.

Mandar Shete, *Principal at Infosys*

Mandar Shete is a Principal with the Infosys Modernization Practice. He has been in the IT industry for over two decades and has played leadership roles in development and maintenance across applications, infrastructure and cloud. Mandar's focus area is building platforms and solutions to help global clients in their modernization journey.

References

[1] https://www.gartner.com/smarterwithgartner/6-best-practices-for-creating-a-container-platform-strategy/

[2] https://landscape.cncf.io/