WHITEPAPER

Cloud-Native IT Transformation: Building Apps Faster and Better

aspire SYSTEMS

attention. always.

www.aspiresys.com

Introduction - Cloud Native Applications

There is no doubt that the fundamental nature of IT in this fast evolving digital world is completely changing from just "supporting the business" to "becoming the business". This transition is pretty much impacting all the key underlying areas of software development such as development and operations, building teams with the right skill sets, technical architectures etc.

Today, the success of a business is determined by the speed with which it can capitalize on new technologies and adjust IT operations.

- Companies like Netflix are deploying thousands of lines of codes every second reducing outages and improving time-to-market.

- After migrating from waterfall delivery model to DevOps, Nordstrom is now able to do monthly releases as compared to bi-annual releases.

At the core of these business transformations, lies an agile and scalable IT platform on cloud, which clearly is the need of the hour – **developing applications at the speed of business.** Migrating workloads to the cloud, adopting agile deployment processes and becoming more responsive to business opportunities are some of the steps representing a starting point for one of the fastest emerging trends:

Cloud-Native IT Transformation. Sounds like another buzzword? It might just be the breakthrough organizations are looking for to speed up innovation.

So far cloud strategies have mostly focused on application migration through a "Lift & Shift" approach. However, high-performing organizations in the new digital economy are those who have unlocked the next level of cloud benefits in terms of agility, innovation and efficiency. The "cloud-native" mindset which includes re-engineering processes, workflows and re-organizing people, is becoming the new normal.

In this whitepaper we will deep dive into the world of cloud-native applications, compare it with the traditional application development, describe the key elements of a cloud-native application and share the secret sauce to build cloud-native apps.

What are Cloud-Native Applications?

Taking Ideas into Production Quickly and Efficiently.

Cloud-native apps are born and raised on the cloud. It's much more than just signing up with a cloud provider and running your applications on cloud. From designing to implementing to deploying, cloud-native applications completely change the entire software development lifecycle by leveraging the advantages of cloud computing models. Cloud-native apps enable building new applications based on cloud principles, using services and adopting processes optimized for the agility and automation of cloud computing. They work on a service-based architecture bringing different components into one place, within a container-centric model. These can be scaled quickly on-demand offering continuous delivery with no interruption to services.

In cloud-native app development, the development and programming are driven by business requirements. Listed below are three key underlying principles of cloud-native app development:

1. **Cloud-first Approach**– Adopting a cloud-first strategy and making the most of all the cloud services and features (cloud infrastructure, networking, security, databases, caching, load balancing etc.) is the first and foremost principle of cloud-native.

2. **Delivery Process Automation** – The second key aspect of cloud-native development is bringing in Agile and DevOps enabling continuous integration and delivery. The primary objective of software development should be to get the minimum viable product to market in the shortest possible time and then quickly deploy incremental releases to continuously improve the overall product.

3. **Scalable Application Architecture** – With the growth of software as a service business, scalable development to support growing user base, managing cost, etc., has become the need of the hour.

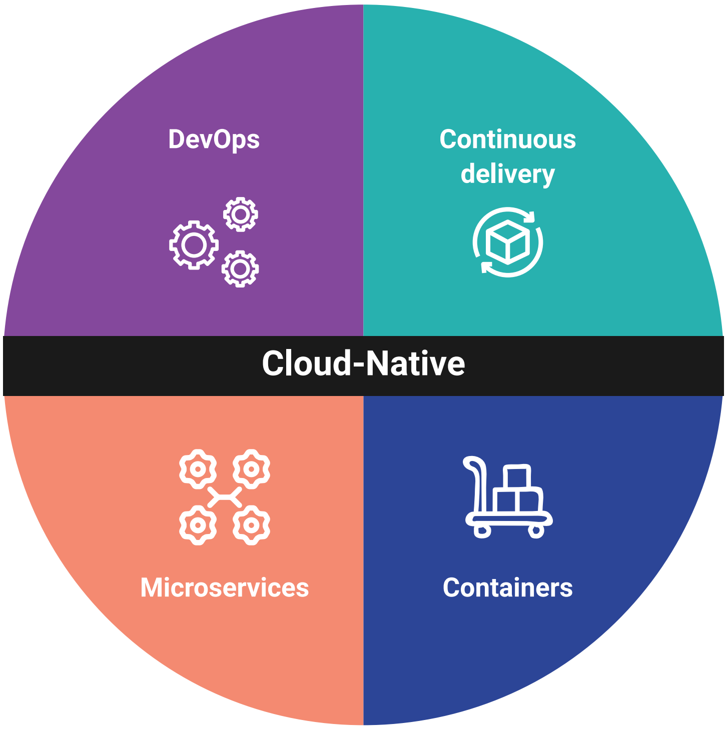
Key Features of an Effevtive Cloud-Native platform

- Easy access to multiple languages & frameworks
- Ability to scale-up/down as per business demands
- Automation powered by DevOps
- Mobil-first solutions
- End-to-end product development and app monitoring services
- API-first approach
- Containers for packaging and versioning
- Hybrid architecture

Four Tenets of Cloud-Native Application Development

Cloud-native is an approach to building and running applications that leverages the advantages of cloud computing delivery model. It implies that the apps live in the cloud, as opposed to an on-premise data center. Cloud-native approach builds applications as microservices and runs them on containerized and dynamically orchestrated platforms. Cloud-native technology empowers organizations to build and run scalable applications on public, private and hybrid clouds. These applications are built from scratch, designed as loosely coupled microservices, optimized for cloud scale and performance, use managed services and take advantage of continuous delivery to achieve reliability and faster time-to-market. The final objective is to improve scalability, speed and eventually, margin.

Cloud-native practices comprise 4 main tenets:



DevOps – Cloud-native changes how developers work and DevOps is all about people and how they work together. In DevOps, the goal is to reduce the time taken by developers to write the code and the time taken for the code to go into production. Cloud-native also has the same goal that is to improve the speed of delivery. Thus, DevOps is one of the first tenets of cloud-native application development.

Continuous Delivery – Continuous delivery is at the heart of the cloud-native movement. Continuous delivery is a set of practices and tools that enables rapid development and testing and frequent release of updates. These architectures typically have many moving services also called Microservices. Each service is independent and stands on its own.

The idea is that an update to a service can be quickly tested with minimal expected impact on other services. Thus, continuous delivery enables applications to release quickly, reliably and frequently, with less risk.

Microservices – The microservices approach is a way of breaking down an application into several modules with well-defined interfaces. Each module performs one business function. These modules are independently deployed and run on their own, communicating over HTTP APIs. Microservices architectures are well-suited for cloud-native applications. An application written following the microservices architecture is automatically cloud-native since this architecture allows the application to be deployed in a distributed manner, which allows it to be scalable and always available.

Containers – Containers and the cloud go hand in hand in the cloud-native world as containers make cloudbased applications easier to deploy. Containers provide light-weight virtualization by dynamically dividing a single server into one or more isolated containers. Containers only contain the code and dependencies needed to run that specific application, making them smaller and faster to run. Moreover, the same container can be deployed in any cloud, so containers provide the flexibility of moving to any cloud provider as they are independent of the underlying infrastructure. Therefore, it helps developers in building cloud-native applications that can run on any cloud, right out of the box. They don't have to rewrite code to run and scale across a range of cloud infrastructures – freeing up the developer's time. As a result, developers can use this time on improvements and new features.

Cloud-Native Application vs Traditional Application Development

Cloud-native application architectures are very different from traditional on-premise enterprise applications.

Here is a list of some of the glaring differences:

Difference	Traditional On-Premise Enterprise Applications	Cloud-Native Enterprise Applications
Programming Language	On-premise software tends to be written in traditional languages, like C/C++, C# or another Visual Studio language if deployed on a Windows Server platform,a and enterprise Java. And if it's on a mainframe, it's likely in Cobol.	Cloud-native applications are more likely to be written in a web-centric language. Such as HTML, CSS, Java, JavaScript, .Net, Co, Node.js, PHP, Python, and Ruby.
Updatability	On-premise apps need updates to be installed separately every time. There is cowntime during the installation of the update.	Cloud-native appas are easy to update and always available. Downtime is not required to update the application.
Elasticity	On-premise applications do not have the option of scaling up resources during a spike. If traffic is higher than the load the app can withstand, the app will crash.	Cloud-Native applications have the advantage of the elasticity of the cloud. During a spike in use, a cloud-native app can adjust to the increased resources and scale as needed.
Multitenancy	Most On-premise apps don't work well in a virtual environment and require a non-virtualized space.	A cloud-native app is able to work in a virtualized space and can share resources with other apps.

Difference	Traditional On-Premise Enterprise Applications	Cloud-Native Enterprise Applications
Connected Resources	An On-primise app is considered to be quite rigid with regards to its connections with the network resources, such as networks, security, permissions, and storage. Most of the resources need to be hard coded, and they are likely to break if any things is changed.	In the cloud, network and storage are completely different. Cloud allows re-platforming. This enables accommodating changes in networking, storage, and even database technologies in an easy manner.
Downtime	On-premise apps may have failover ready but in most cases if the server goes down, the app goes down with it.	If a cloud providers suffers an outage in one location, another location will pick up the stack and ensure the app is always available.
App Management	On-premise apps have to be managed manually.	App management in the cloud is mostly automated.
Modular Design	On-premise apps tend to be monolithic in design. It is usually one big app with a whole lot of subroutines.	Clioud-native apps are much more modular if compared to the traditional on-premise. Many functions are broken down into microservices. This enables them to be shut off when not needed and for updates to be rolled out to that one module, rather than the full application.
Statelessness	Most on-premise apps are stateful. they store the state of the app on the infrastructure on which the code runs. This allows the app to break when adding server resources.	Cloud-native apps are not really tied to any infrastructure which means they are stateless.

Critical Success Factors for Building Cloud Native Applications

Navigating the Cloud-Native Journey Seamlessly.

1. **Create a DevOps Culture** – The first and foremost step is to have a DevOps mindset and incorporate an agile approach for building and deploying apps faster. Quality assurance, testing and security teams need to be involved right from the requirement stage. Implementing DevOps process is just not about tools and technologies, but also collaborating across teams to create an integrated approach to developing and delivering applications faster. In a typical cloud-native environment powered by DevOps, teams leverage open source technologies rapidly, build prototypes and adopt agile workflows.

2. **Provide Access to Automation and All the Tools** – One needs to provide accessibility to all the tools and automation that the developers need. You need to enable them so that you are able to increase the DevOps capabilities and commence training. This helps in optimizing efficiency and truly maximizing product velocity.

3. **Use Modern Architectures** – Instead of using old architecture patterns, you need to make use of modern architectures which are up to date. Cloud-native essentially, means using microservices and a reactive type of architecture. Rearchitecting also helps as the organization then, can take advantage of a cloud-native architecture. When embracing cloud-native, companies need to realize that it requires a completely different culture to take advantage of the cloud and automation.

4. **De-risk Infrastructure by Stepping to the Cloud –** These cloud-native strategies are known to fundamentally shift infrastructure planning from high to low risk. It also, significantly reduces the operational load for the company's technical teams. The transformation provides fast access to flexible, virtual resources that scale effortlessly at need. By moving the infrastructure to cloud, you pay only for actual resources consumed and cloud mitigates every major traditional infrastructurerelated risk.

5. **Create a More 'Service-oriented' Organization –** You need to organize your teams around specific services or capabilities. This particular change needs to be very carefully managed in order to avoid any kind of resistance. A 'service-oriented' enterprise facilitates cloud-native computing which further, takes advantage of many modern techniques, including PaaS, Multi-cloud, Microservices, Agile methodology, Containers, CI/CD, and DevOps.

6. **Self-service Agile Infrastructure** – Another term for self-service agile infrastructure is platformas-a-service. When you use a platform-as-a-service, all you need to do is hand the platform your application code and it takes care of the rest. The self-service agile infrastructure is responsible for providing your application with the backing services it needs to run.

7. **Choose the Right Tools and Speed Up Existing Application** – Cloud-native development requires choosing the right tool for the right task. The platform must offer the right mix of frameworks, languages, and architectures to support the chosen development requirements, hence, it is essential to choose the right tools. Similarly, when embarking on a cloud-native application journey, organizations should not only focus on new development but also, take a fast monolith approach by moving the existing monolithic architecture to a more modular, service-based architecture and API-based communication.

8. **Automation for Accelerating Application Delivery** – IT automation is essential to accelerating the delivery of cloud-native applications by elimination of the manual tasks. IT automation tools create processes, rules, and frameworks that can replace or reduce labor-intensive human interaction that delay time to market. Automation is key to IT optimization and digital transformation.

9. **Use Application Services to Facilitate Development** – Cloud-native applications require one or more of the application services – Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS) or latest paradigm of serverless application, Function-as-aService (FaaS), to help developers accelerate development and get new applications to market faster. Application services accelerate the development of the cloud-native application.

10. **Provide Advanced Deployment Techniques and Accelerate Continuous Delivery** – Providing advanced deployment techniques helps in reducing the risk of software releases and it builds an environment for experimentation. It is essential for facilitating innovation across an enterprise. Agile development methods release early, release often. DevOps and continuous delivery approaches unites developers, operations, quality assurance, and security teams to improve software delivery processes thereby, accelerating continuous delivery.

Business Benefits - Why Cloud-Native Apps Matter?

Most enterprises have started building their applications strategies and roadmap on cloud platforms. Cloudnative is expected to gain more prominence and popularity in the near future. Cloud-native applications are redefining the way enterprises scale their businesses. Tech-savvy enterprises are ready to deploy gamechanging technology solutions, and cloud-native is able to help them stay ahead of the competition.

Here are some of the major reasons why developing cloud-native applications is advantageous and worth the time and money:

1. **Auto-provisioning and Auto-scalability** – In case of traditional applications, the resources are physically provided but cloud-native applications enable automatic provision of resources and manage resources automatically. They facilitate on-demand, self-service, programmatic provisioning, and releasing of resources used for compute & storage services. Cloud-native applications are also enabled with auto-scale feature and hence, can handle continuous business needs automatically. This feature is essential as it assists organizations to deal with complex process requirements with the pay-as-you-go model.

2. **Minimizing Failure Risks** – Cloud-native applications are inherently resilient to failures because of their auto-redundancy feature. They can automatically, handle the outages and enable corrective actions and in the event of failure, application processing instantly moves from one data center to another without interrupting the service. Also, in case of occurrence of a partial outage in any one data center, the cloud-based applications will still continue running seamlessly.

3. **Reduced Cost and Ease of Management** – Cloud-native has several options to make infrastructure management effortless. Serverless computing platforms take care of configuring networking, provisioning cloud instances and allocation of sufficient storage. Enhanced cloud-native capabilities such as Serverless enables you to run dynamic workloads and utilize pay-per-use compute time in milliseconds. Ultimate flexibility in pricing has been enabled by cloud-native.

4. **Building More Reliable Systems** – With the modern cloud-native approaches, one can easily build applications which turn out to be fault-tolerant with resilience and self-healing built in. Cloud -native microservices helps you to achieve higher uptime and thus, improve the user experience.

5. **Decreased Time-to-market and Security** – IT operations time is significantly reduced by effectively transforming the decisions about provisioning, scaling, continuous integration, regression testing, and zero-downtime deployment into automated tasks. Cloud-native application patterns are based on loosely coupled microservices, which help in greatly reducing the operational and security risk of massive failures.

6. **Greater Mobility and Reduced Downtime** – With a well-orchestrated DevOps mechanism, you are able to promote better communication and collaboration thereby, maintaining service stability and minimizing downtime. With an API-first approach, you also, have much greater mobility and hence, you can emphasize where it is most required- mobile, the web, or extending off-the-shelf applications.

The advantages of cloud-native applications help enterprises to improve their business processes with minimized infrastructure overhead, costs, and manual efforts. These applications are highly resilient, scalable, portable, and can be updated in an hassle-free manner. This is the best time to realize the full potential of the cloud by re-architecting or rebuilding your applications as cloud-native. Think 'cloud-native' and begin your modernization and transformation journey.

Cloud-Native: The Future of Application Development

It’s time to make your move!

Cloud-native is currently considered to be one of the biggest trends in the software industry. It has already changed the way people think about developing, deploying and operating software products. It is a lot more than mere association with a cloud provider and using it to run your existing applications. It considerably, affects the design, implementation, deployment, and operation of your applications. Cloud-native technologies and applications are growing and can play an integral role in how companies transform themselves. The ideas and concepts of cloud-native computing have introduced a new way to implement complex, scalable systems. Cloud-native computing is not only a competitive advantage; it is also about keeping up and ultimately moving ahead of your competitors. This trend will only continue and, as the ecosystem around cloud-native technologies grows, it will become the competitive differentiator that will keep your company relevant in the future. To remain competitive and keep ahead of the competition, you must adopt it.

Practice Head

Aju Mathew,
Vice President – Development Services (Enterprise & ISVs)

Aju is the head of the Software Engineering service line at Aspire Systems with an industry experience of over 22 years specializing in ERP, Portal, Integration and BI solutions. He has helped transform an initial business from just a C/C++ team into a cross technology skilled team and an Enterprise solutions service line into a 1200+ member team specializing in 4 practices.

If you'd like to know more about our Cloud-native development services, please write to us at **info@aspiresys.com**

About Aspire

Aspire Systems is a global technology services firm serving as a trusted technology partner for our customers. We work with some of the world's most innovative enterprises and independent software vendors, helping them leverage technology and outsourcing in our specific areas of expertise. Our core philosophy of "Attention. Always." communicates our belief in lavishing care and attention on our customer and employees.

For more info contact: info@aspiresys.com or visit www.aspiresys.com

USA	+ 1 630 368 0970
SINGAPORE	+65 3163 3050
INDIA	+91 44 6740 4000
BELGIUM	+ 32 3 204 1942
NETHERLANDS	+ 31 (0)30 800 92 16
POLAND	+48 58 732 77 71
MEXICO	+52 222 980 0115