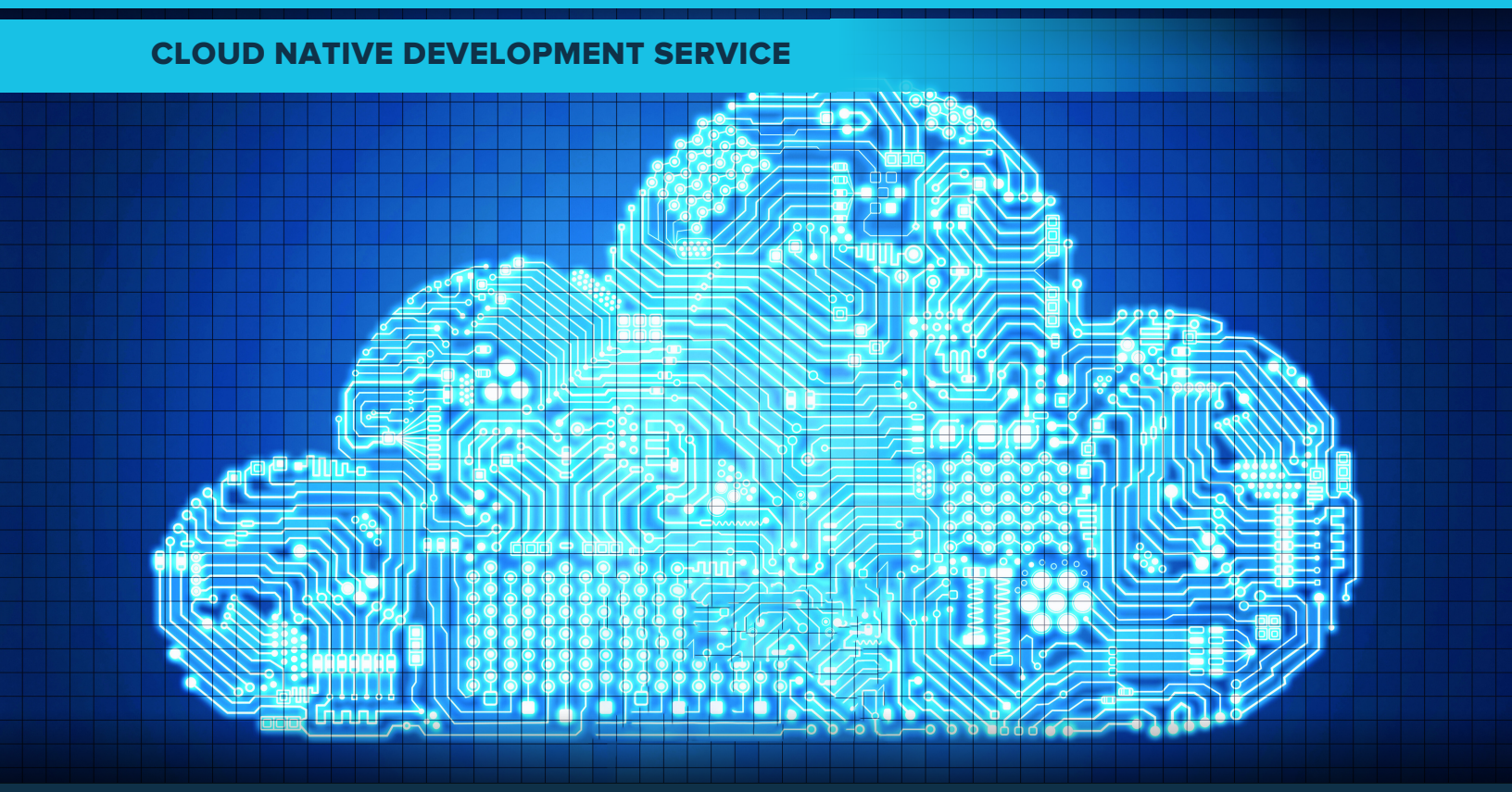
JUNE 2023

GROWTH ACCELERATION PARTNERS

THE BUILDING BLOCKS OF CLOUD-NATIVE: FROM SERVERLESS TO IaC

Now is the Right Time for Your Org to Go Cloud-Native — But First, Learn What Mistakes to Avoid

CLOUD NATIVE DEVELOPMENT SERVICE



The road to becoming a cloud-native organization is a complex one — and it's not getting any simpler. Amazon Web Services (AWS), at last count, offers 238 services across the cloud development pipeline, while Microsoft Azure's total number of services is also north of 200. Rest assured that these numbers will keep going up, and if you're not yet focusing on cloud-native applications, you will fall behind.

WeAreGAP.com

It's hard to know where to start, so let's start at the beginning. Yet even a simple definition offers more questions than answers. Ask half a dozen people to define cloud-native, and you'll get half a dozen definitions. This is proven in market research; a 2021 study of 1,000 developers and IT decision makers saw 41.7% define it as writing applications that specifically leveraged underlying cloud infrastructure; 34.5% believed it included applications that utilize Kubernetes and containers; and 23.8% defined it more broadly as moving to a cloud infrastructure provider. These definitions can all be seen as correct in their own way, but Growth Acceleration Partners (GAP)'s point of view is that simply using cloud infrastructure or using containers is only scratching the surface.

To be truly cloud native — and to achieve many of the benefits — you need to leverage cloud technologies.

When it comes to the "why" for cloud-native, the number one answer is usually speed, from faster application building, to more agile frameworks, to quicker time to market. Capgemini places it as one of the three most important factors, along with cost efficiency and customer experience.

A truly cloud-native application is one that provides a consistent experience across all environments, from public, to private, to hybrid cloud. There are two main steps:

FIrst:

Create a virtualized environment for your application that can be executed anywhere. Containers are a clear example of such a "build once, run anywhere" ethos.

After that:

You must be involved in how your application will be executed in the cloud — and this is where the headaches can come in.

These are your biggest challenges, but also the biggest opportunities. There are emerging paradigms and architectures that can make things easier. Serverless architectures are where a cloud provider is responsible for managing both infrastructure and application scaling. Compared to traditional IaaS (infrastructure as a service) models, where the user is responsible to scale up server capacity, a serverless model — using an event-driven architecture — means an event triggers application code to run, and resources are dynamically allocated by the cloud provider.

The goal is to lessen time spent scaling applications and provisioning servers.

With infrastructure as code (IaC), the goal is to save time and resources in managing, monitoring and provisioning resources. Think of a traditional desktop application, which is a single installation. If you had a similar application in the cloud, which is considered a single system and managed as a single unit, it would provision and connect the resources to deploy that system. Tools such as Terraform make it possible to manage infrastructure for applications across multiple cloud providers using a single tool. The ideal level of abstraction for this paradigm would be to install the same application in AWS and Azure — but we are a long way away from that right now.

Also, infrastructure automation powered by GAPBuilt Accelerators can create faster "time to value" by saving businesses six to nine months on engineering projects. The proprietary framework is powered by Terraform and enables GAP clients to deploy or destroy infrastructure at the push of a button, saving them time and resources during a development project. GAP's accelerator simplifies app development, data ingestion and modern analytics with reusable architecture and infrastructure components.

If you have decided cloud-native is the way to go, there are many roadblocks and pitfalls to avoid along the way. Above anything else, cloud-native development requires a significant shift in mindset. Historically, developers have focused on writing the code, be it on Ruby on Rails, .NET, CSS or JavaScript, with someone else then deploying it into a server. Getting out of this mindset can be difficult. After all, if your organization's infrastructure has been on-premises for the past decade or more, then a cloud transformation does not happen overnight.

GROWTH ACCELERATION PARTNERS

To give a simple idea of the variety and challenge of building yourself, let's say you want to install an application in AWS. You can:

- Do it yourself with EC2
- Go the containerized route with AWS App Runner, Amazon EKS (Elastic Kubernetes Service) or Amazon ECS (Elastic Container Service)
- Create your own virtual server
- Run the application or container via Lambda Functions
- ... and so on.

Azure, by comparison, is no less headache-inducing. It's not so much a matter of getting it done, but getting it right.

Don't make the mistake either of thinking you're going cloud-native when instead, you're moving your existing workload to the cloud — as it could be a very expensive error. Respondents to Flexera's 2023 State of the Cloud survey estimate more than a quarter (28%) of their cloud spend to be wasted. Again, AWS has introduced products such as Cost Anomaly Detection to help in this regard, but there is a link between inflated cloud spending and thinking "lift and shift" is the extent of the project.

GAP has a team of cloud-certified architects who have been building infrastructure for 7+ years; these data engineers and data scientists have been building ETL pipelines and insightful real time dashboards for some very large customers. Therefore, working with a company such as GAP means you get a set of expert services depending on where you are in your journey:

- Developing an application without going 100% cloud-native if preferred — though not recommended;
- Helping with development and deployment in a way that can be easy to handle; or
- Creating a fully cloud-native solution using the best technologies available.

Also, it's important to keep in mind that even if you have a highly-technical team, those skills may not translate to cloud-oriented development. One client GAP recently worked with was highly-technical, but had limited experience in cloud — and the recommendation for modernizing their existing platform was to move forward with a cloud migration.

This best technologies approach can sometimes mean mixing and matching between different provider offerings. A good example is having your infrastructure on AWS, and your data analytics on Snowflake. The companies are partners, but AWS has its own data warehouse offering in Amazon Redshift. With GAP as your partner, you will have assurance that the implementation is done right technically — as specific GAPBuilt Accelerators, which deploy IaC to save engineering and development time, include AWS and Snowflake capability — plus you'll receive vendor-neutral expert advice and details related to best practices.

Moving to cloud-native gives you various tangible benefits, from cost efficiency and faster time-to-market, to an improved customer experience. Lift and shift is not going to be enough. With the complexity of public cloud providers making a precise roadmap difficult, the change in mindset required to go from legacy to cloud-native, and the emergence of IaC and serverless technologies, partnering with a company like GAP will provide the technical skills, the soft skills, and the framework for proven approaches to meet your business objectives.

Cloud-native development is tough. Contact us to see why a partnership with GAP makes sense as you move forward.

To find out more, please visit WeAreGAP.com

company/growth-acceleration-partners/ @GrowthAccelerationPartners @GAPapps