

# FROM INFRASTRUCTURE TO APPLICATIONS —

## Once You're Cloud Automated, You'll Wonder Why You Weren't Before

### CLOUD AUTOMATION

If you use cloud automation to streamline your infrastructure, you'll be forgiven for wondering how you ever lived without it. And if you have not already taken the plunge, now is the time to do so!

According to a 2023 "State of IT Automation Report," cloud automation topped the list of planned automation tool investments, and more enterprises are running permanent product jobs in the cloud than test or ad-hoc development jobs.

---

Cloud automation can broadly be defined as **implementing tools to reduce the effort of manual processes** in provisioning, configuring and managing cloud environments, be they public, private, hybrid or multi-cloud. Previously, organizations could only control their infrastructure simply by committing more personnel and money to it, hiring engineers to create live backups and monitor CPU utilization among others.

---

Cloud infrastructure automation tools take various forms. It can include everything from using Chef, Puppet or Ansible for configuration management — installing and managing software on existing servers — to AWS CloudFormation and Terraform, which are primarily provisioning tools, focused on provisioning the servers, as well as other infrastructure. Yet that is not all.

---

Kubernetes is another example of a technology that focuses on automating the deployment, scaling and management of containerized applications. Also, serverless technologies, from AWS Lambda to Microsoft Azure Functions, have degrees of automation through their event-driven models, helping run scheduled jobs to clean up infrastructure or other repetitive tasks.

---

In various examples, you can see how more than one tool can be used. For instance, you can use Terraform for deploying all the underlying infrastructure, and Ansible to deploy applications on top of that infrastructure. You define the resources you want to create, from VMs, to networks, to security, with Terraform code — otherwise known as infrastructure as code (IaC) — before running Ansible playbooks, to manage the configuration of your infrastructure, on the resources.



The benefits of cloud automation are evident if utilized correctly, from faster deployment, to clearer control over your IT processes and policies, and lower costs. Yet whether you are an Amazon Web Services (AWS) or Microsoft Azure organization, the varied possibilities need an element of nuance to avoid going down the wrong path.



**What is your use case?**



**Do you want to automate workloads, the infrastructure itself, or both?**



**Where do you want your code to execute?**



**Do you need to bake in an abstraction layer?**

---

**To best answer these questions, you will find benefits from working with a trusted partner like GAP.**

---

Many organizations will understand a simple “lift and shift” for their monolithic application will not be the most effective or economical. GAP experts can help with both the consulting — what can be containerized, what can be moved into a serverless architecture, etc. — and the deployment. We can recommend what efficiencies can be brought about through automation for a client’s application or website.

Some efficiencies are strikingly obvious. One client was a GAP customer in the financial sector, providing software to banks to help them assess financial risks. The software needed to scan millions of records from sanctions lists and provide record matches against other lists. Yet it simply could not keep up with demand, being built on the old .NET framework and hosted on-premises.



**The solution proposed by GAP was an architecture based on various Azure serverless offerings to scale in and out based on load. The products used were Azure Functions and Azure Durable Functions, along with Azure Service Bus.**

Azure Functions, as a serverless solution, automates the process of deploying and maintaining servers, with the cloud infrastructure providing all the up-to-date resources needed to keep applications running. Azure Durable Functions is an extension that enables the writing of stateful functions, meaning it retains memory of previous calls or requests, automating state management and checkpoints, and providing further efficiency. Part of the process involved refactoring, as well as using the newer .NET 6 framework, to deploy things to Azure.

While this was working well, further down the line, with GAP's assistance, the client changed its architecture from containers to more of a serverless configuration. This involved the use of a few products, from AWS Lambda, to Amazon Simple Queue Service (Amazon SQS), which lets you send, store and receive messages between software components without needing to manage software or maintain infrastructure. This

evolved into using AWS SAM (Serverless Application Model), which provides templates optimized for defining IaC for serverless applications. A key benefit of this was a much more easily manageable learning curve for the client's development team.

These client examples showcase both the breadth and depth of GAP's capabilities, not just in terms of use case or cloud provider, but from which stage the

client is at on its journey, and which processes need automating.

Pretty much anything is fair game to be automated, from cloud infrastructure and SaaS tools, to cloud-native applications. More than two in five (42%) organizations plan to invest in cloud automation in the coming year. If you think you should be among them, let GAP show you what you can do.



01022024

To find out more, please visit [WeAreGAP.com](https://www.WeAreGAP.com) 