2024

CLOUDFUEL WHITEPAPER

Deep dive into our newest cases

Cloud Fuel C

– INTRODUCTION

Dear reader,

Welcome to our CloudFuel whitepaper, where we introduce 7 recent case studies that showcase our expertise in cloud migration and modernisation.

Our cases span a range of industries and challenges, from ISV container migration to Azure Kubernetes Service (AKS), to healthcare application migration to Azure Container Apps (ACA), to Jira to Azure DevOps migration and development coaching for a cloud-native application.

Each story presents unique challenges and our innovative solutions, demonstrating our ability to provide tailored and effective support to our clients. Our services encompass the full spectrum from initial assessment to implementation, ongoing support, and knowledge transfer, based on the specific requirements of our clients.

Join us as we explore these case studies in detail and discover how CloudFuel can help businesses like yours navigate the complexities of cloud migration and modernisation. Your journey awaits, and we look forward to the possibility of collaborating with you.

— TABLE OF CONTENTS

ISV Container Migration to AKS: **Smart CV matching tool hosted on Azure.** 4

Healthcare Application Migration to ACA: **Optimizing performance and maintenance of a digital platform.** 12

High Security Cloud Migration Support: **Expert guidance on application modernization to a cloud-native architecture.** 18

Jira to Azure DevOps Migration: **Reshaping IT environment for a reusable plastic packaging provider.** 24

Healthcare Cloud Migration Support: **Moving on-site infrastructure to the cloud for a hospital.** 31

Azure Migration and modernisation Program Support: **Moving existing applications onto Azure for a water utilities provider.** 38

Development Coaching for a Cloud-native Application: **Expert coaching for the creation of a standalone cloud-first application.** 45

CONTAINER MIGRATION TO AKS

In this case, we tackled a unique challenge for a software company. They created a smart CV matching application, and their client needed it hosted on Azure. In this story, we'll see how CloudFuel's know-how worked hand in hand with our client's existing technology to create a successful plan. We'll explore the blueprint for their Azure setup and learn about the smart choices made for performance, security, and saving money.

– THE CHALLENGE

Azure AKS Deployment

The challenge was twofold and intriguing. Our client, an Independent Software Vendor (ISV), needed to establish a **dedicated tenant** for one of their own clients, distinct from their default setup. This deployment had to be **scalable**, easily **maintainable**, and fortified with robust **security measures**. Ensuring that the new Azure deployment met these criteria was a mission that demanded a harmonious blend of innovation and practicality.

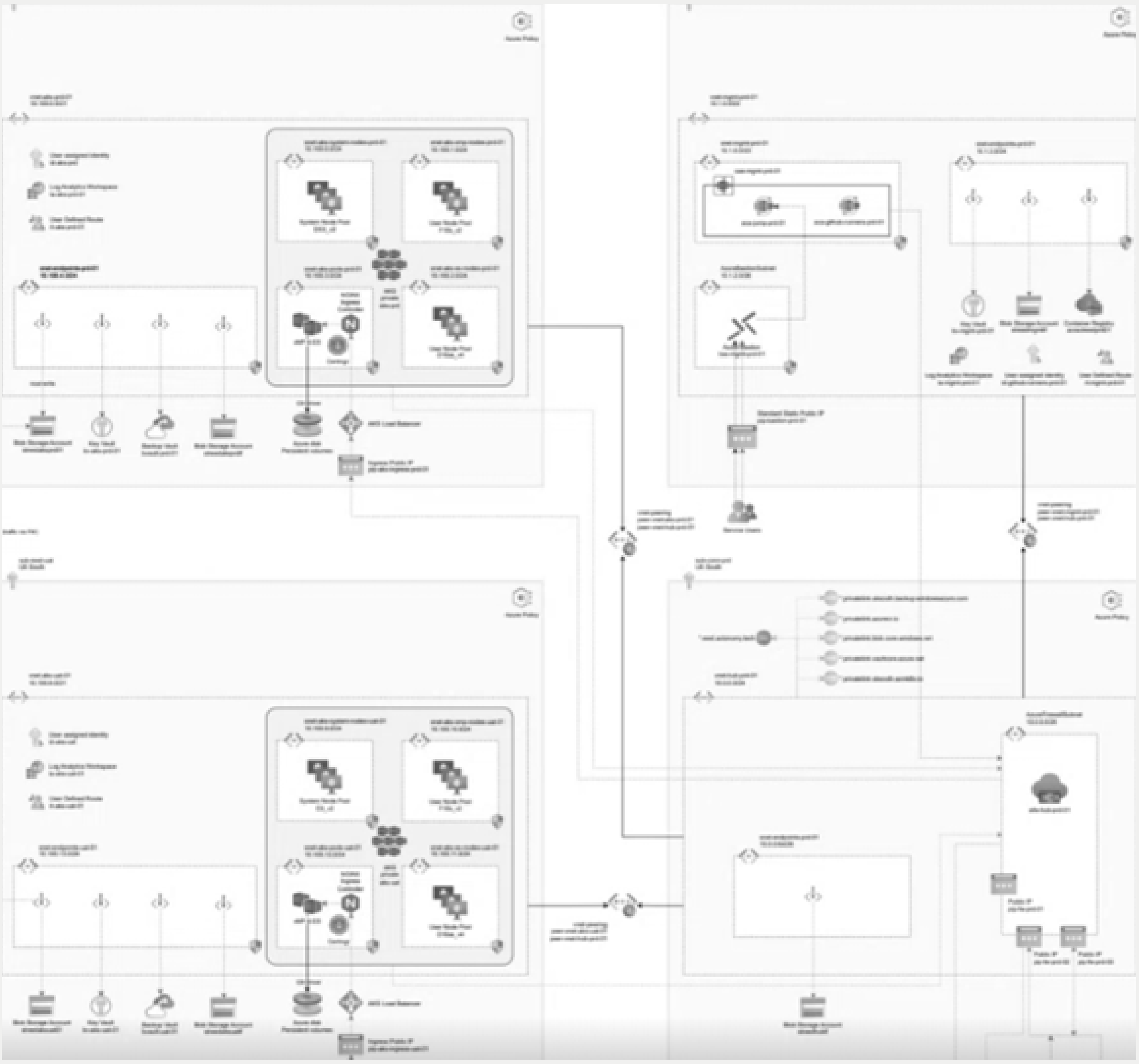
– OUR SOLUTION

Blueprint for Success

We decided to deploy the client's existing container-based architecture with **Azure Kubernetes Service (AKS)** instead of Azure Container Apps (ACA), because they needed a large amount of computing power from time to time.

However, it's important to note that hosting application containers is just one piece of the puzzle in creating a fully functional cloud deployment. Our initial task was to craft a **reference architecture**, which would later serve as the blueprint for the client's Azure infrastructure.

While we can’t reveal all the intricate details here, we added a **low-res version** of our architecture below. We’re sure an experienced cloud architect will already be able to learn much from this low-res version. Certainly give us a call if that’s you, but even for the rest it will serve as a **reference point** for the rest of our discussion.



Four key squares

The four prominent **yellow areas** represent **subscriptions**, primarily used for effective budget management. Each of them encompasses a logical grouping of components. Starting from the bottom left and moving clockwise, we have a testing environment, the production environment, the management subscription, and finally, the connectivity subscription.

Zooming in on the squares surrounding the **purple honeycomb**, we delve into the heart of the infrastructure: the **application containers**. Notably, the client's xMP application relies on Elasticsearch. To strike a balance between performance and cost-effectiveness, we chose CPU-heavy nodes for the xMP matching engine while maintaining a 1:1 CPU/memory ratio for the elastic nodes. We also made sure to add a **system node pool** and **user node pools** for xMP (CPU-based) and Elasticsearch.

Secure deployment

Each node pool resides in a **distinct subnet** to facilitate the management of incoming and outgoing traffic and ensure that only the necessities are admitted. An intriguing aspect of this setup is the use of the **NGINX Ingress controller** in combination with **Cert-Manager**, which employs Let's Encrypt as the issuer. This setup offers several advantages:

Automatic certificate management: Cert-Manager automates the process of requesting, renewing, and managing certificates, eliminating concerns about certificate expiration.

Simplified configuration: Certificate and issuer configurations are defined in Kubernetes resources, simplifying management within your infrastructure.

Scalability: This setup scales seamlessly with your Kubernetes cluster and can be employed for multiple applications and domains without added complexity.

In terms of security, the deployment involves **private configurations** with all traffic routed through a firewall. **HTTPS** is enforced, making port 80 inaccessible. This presented an interesting challenge as Let's Encrypt typically uses an http-01 challenge, which relies on port 80. To address this, we configured Cert-Manager to work with a **public DNS subnet**, allowing the use of the Let's Encrypt dns-01 challenge. While it may sound complex, the extra configuration enhances security by blocking HTTP traffic.

Cost-effective scaling

Implementing a jump server for remote CLI access and managing our self-hosted GitHub runner for deployments, we needed a solution that didn't require continuous active hardware. This was efficiently achieved by leveraging the scale-to-zero feature of Azure Container Apps. The transition of the jump server from a virtual machine to this platform resulted in significant cost reductions, with monthly expenses plummeting from approximately €50-80 to as low as 50 cents, depending on usage.

CloudFuel deployed xMP on Azure with **AKS for performance,** cost-effectiveness, and security. Achieved savings with robust functionality and scalability.

– THE RESULTS

In addressing our client's challenge of hosting their intelligent CV matching application xMP entirely on Azure for their client, CloudFuel leveraged its expertise in modern cloud systems to provide a well-thought-out solution. The deployment on Azure Kubernetes Service (AKS) was a natural fit, and our meticulous approach to crafting a reference architecture ensured that the client's Azure infrastructure was built on a solid foundation.

The four key subscription areas and the intricate details of application container management were strategically designed to optimize performance and cost-effectiveness. Security measures, including private configurations, a firewall, and enforced HTTPS, were implemented to safeguard the deployment. Cost-effective scaling was achieved, resulting in significant savings while maintaining essential functionalities.

HEALTHCARE APPLICATION MIGRATION TO ACA

In this case, we'll explore how our client, a hospital that's at the forefront of utilizing new technology in healthcare, navigated the journey of enhancing their digital platform, initially hosted on Amazon's cloud infrastructure. To optimize performance and ease of maintenance, the hospital decided to migrate all their applications onto Azure, a more maintenance-friendly environment. Join us as we share their story of this strategic migration, backed by the unrivaled expertise of cloud migration specialists at CloudFuel and the advantages of Azure Container Apps (ACA).

– THE CHALLENGE

The digital platform of our client was originally crafted using Java and ran efficiently on Amazon's cloud infrastructure. But as they moved towards optimizing the performance and upkeep of their platform, a plan was drawn up to shift their application to a more **maintenance-friendly environment**. This planning culminated in a strategic decision to migrate all their applications onto Azure. After in-depth strategy discussions with CloudFuel, the cloud migration experts, **Azure Container Apps (ACA)** was selected for its superior maintenance advantages.

– THE CLOUDFUEL SOLUTION

The hospital decided on Azure Container Apps (ACA) due to its promises of a maintenance-friendly solution. However, it was **new terrain**, and unknown at the time, ACA did not yet support User Defined Routing (UDR), a feature essential for creating secure network routes.

The **absence of default UDR support** presented the first challenge. But CloudFuel's team took it in stride, using their expertise and innovative problem-solving skills to work around the obstacle. Now, UDR is a default feature in Azure Container Apps —but at the time, the ingenuity of CloudFuel's team was instrumental in overcoming this hurdle.

Next, there was a need to set up Continuous Integration and **Continuous Deployment (CI/CD) pipelines** for automatic application updates. This process allowed efficiency and ensured that the latest security measures and functionality improvements reached the patients at all times. Furthermore, CloudFuel adeptly utilized its skills to implement Terraform efficiently for ensuring **Infrastructure as Code (IaC)**.

Another crucial issue was the **single port per container** that was initially supported. CloudFuel took this as yet another challenge to overcome—being flexible and adjustable to incorporate support for multiple ports when Azure previewed this functionality.

Finally, CloudFuel also implemented a **Customer Managed Key (CMK)** customer key rotation in our client's digital platform. This is a security best practice in the management of cryptographic keys utilized in cloud environments. By managing these encryption keys, the security of stored data is significantly enhanced, reducing the risk of unauthorized access.

With regular key rotation woven into the CI/CD pipeline, CloudFuel not only ensured automatic, secure updates of the application but also made certain that the security of the data maintained its integrity with each update. Thus, key rotation paved the way for a more secure, robust and reliable digital healthcare platform, adhering to the best practices in information security.

CloudFuel's innovation ensures **secure, high-performance healthcare platforms.** Terraform integration ensures adaptable infrastructure, reinforcing our commitment to continuous innovation and superior patient care.

– THE RESULTS

Thanks to CloudFuel's innovative approach, our client's digital healthcare platform runs smoothly and securely, offering unparalleled performance. The introduction of Terraform ensures the infrastructure remains adaptable to changing requirements.

These results underpin the incentive for continuous innovation and a proactive approach in the cloud services domain, proving that with the right knowledge, even the most complex challenges can be overcome. Leveraging CloudFuel's expertise has enabled our client to continue delivering superior patient care via its digital platform, staying true to its vision even amid technical complexities.

HIGH SECURITY CLOUD MIGRATION SUPPORT

At CloudFuel, we specialize in guiding companies on application modernisation journeys to a cloud-native architecture. In this case study, we'll look at how we helped our client with the migration of their microservice architecture with expert advice, dedicated workshops, and hands-on knowledge transfer. Let's dive in.

– THE CHALLENGE

Our client, a player in electrical equipment distribution, stood at a pivotal juncture. Their **Java microservices**, the backbone of their operations, were ready to make their **transition to the cloud**. Additionally, they recognized the significance of adopting a **private network architecture** as part of a robust security strategy to safeguard their digital infrastructure in the cloud. With a proficient operations team already in place, they sought the right expertise to steer their migration.

– THE SOLUTION

Part 1 Finding the right fit

Our collaboration commenced with a **workshop**. These sessions went beyond presentations; they were strategic discussions aimed at pinpointing the **most fitting cloud solutions**. Guided by CloudFuel's expertise, the workshops navigated through an array of **Azure compute solutions**, each offering distinct advantages. However, the exploration quickly converged on an obvious candidate: Azure Container Apps (ACA).

ACA's **managed environment**, purpose-built for microservice containers, offered a solution where simplicity met scalability. The promise of **serverless architecture** and **automated** scaling resonated with their vision of seamless operations. It ensured that our client's teams could concentrate on crafting code, leaving most of the infrastructure management to ACA.

Part 2 Secure transition

At the core of CloudFuel's strategy was a strong emphasis on **knowledge transfer**, perfectly aligning with our client's imperative for infusing expertise into their existing teams. This partnership went beyond mere migration—it was a **transformative effort** to equip our client with the precise knowledge needed to navigate the complexities of the cloud landscape with confidence.

To complement our client's security approach, firmly rooted in a **zero-trust** and **defence-in-depth** strategy, the decision was made to adopt a private network model. Leveraging Azure Virtual Networks (VNets) and strategically segmented subnets, this approach was carefully crafted to enhance the integrity of their data.

In this context, CloudFuel strategically identified **three key microservices**, each with minimal external dependencies. These microservices were positioned within private subnets, forming an invaluable **testbed**. This environment provided a perfect environment for our client's teams to learn and grow, assisted by CloudFuel's expert **mentorship**. This approach has now paved the way for seamless upcoming migrations, ensuring both technical prowess and optimal support.

Part 3

Self-hosted agents for DevOps pipelines for devops

Amidst the cloud evolution, the crown jewel was the deployment of **self-hosted agents** for Azure DevOps pipelines on ACA, which offer a **significant boost in security and control** over the software lifecycle.

Because the Azure DevOps pipelines use our client's private self-hosted agents, the application's **Azure resources are only exposed during deployments**. This comes in handy when deploying to isolated back-end resources. Self-hosted agents reduce the risk of third-party actors accessing the Azure resources (with or without consent) compared to Microsoft-hosted agents, since these are prone to certain secrets that may circulate on the internet.

Moreover, self-hosted agents provide an avenue for **customization**. Our client can tailor the CI/CD pipeline to meet their unique requirements, integrating security measures, compliance checks, and quality gates into every step. This level of customization ensures that the **software lifecycle** aligns seamlessly with our client's internal policies and industry regulations.

The digital transformation journey leveraged technology, achieving:

- Microservices via Azure Container Apps for scalability and simplicity.
- Enhanced security with self-hosted agents.
- Knowledge transfer for future readiness.

– THE RESULTS

Embarking on a digital transformation journey, our client harnessed the power of technology to reshape their landscape. The results spoke for themselves:

Microservices with Azure Container Apps (ACA): ACA seamlessly merged with our client's microservices setup, blending simplicity with scalability. This integration accelerated new feature launches and streamlined operations.

Enhanced Security and Flexibility with Azure DevOps self-hosted agents: Self-hosted agents beefed up security by keeping data within our client's control. This customization ensured alignment with internal policies and industry norms, bolstering resilience.

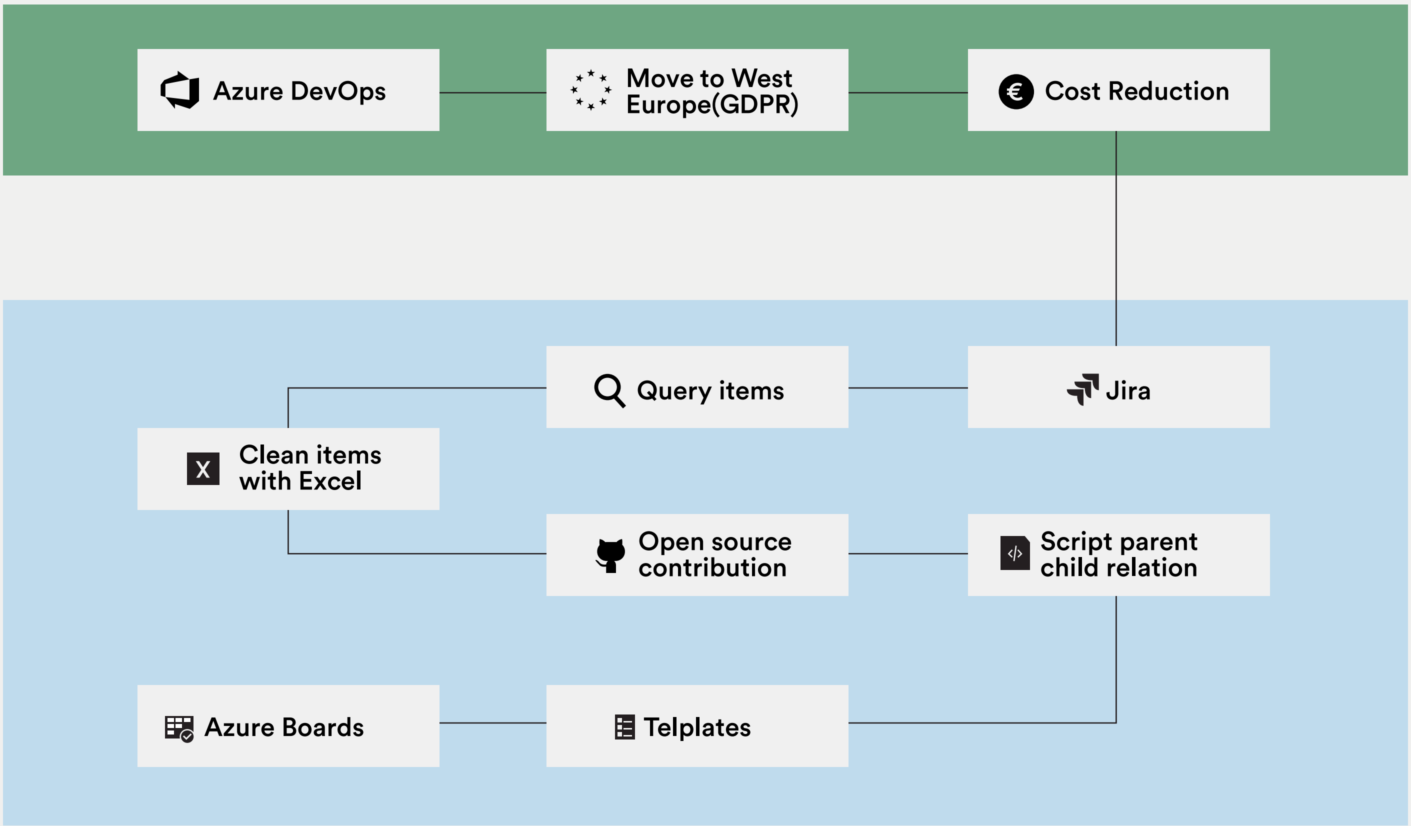
Knowledge Transfer and Future Readiness: Beyond tech, our client received hands-on insights from CloudFuel's experts during migration and deployment, readying them for future autonomy.

JIRA TO AZURE DEVOPS MIGRATION

The next case study is about our collaboration with a provider of reusable plastic packaging. Together, we embarked on a mission to reshape their IT environment. Our journey involved migrating from Jira to Azure DevOps, optimizing processes, reducing costs, and navigating the intricate GDPR compliance landscape.

– THE CHALLENGE

As a reusable packaging provider, our client relies on an intricate IT setup to handle communication with its clients. Relying on Jira for issue tracking, multiple projects were spread across the platform, leading to an environment that could improve the structuring of the existing Jira issues. Different projects and fields added to the complexity, making efficient management a daunting task.



– THE SOLUTION

A Masterful Migration

The pivotal moment came when our client's EMEA office received a **directive** from its American head office to transition from Jira to Azure DevOps. This directive included the provision of an **Azure DevOps environment and licenses**. However, upon closer inspection, we identified a critical issue. All the licenses provided were of the highest tier and tied to a US region tenant, which raised significant **GDPR concerns**.

The first step in this technological transformation was therefore migrating existing projects from the US organization to EMEA. To do this efficiently, we utilized the **Microsoft Process Migrator**. This tool was necessary because some process changes were made; without it, we wouldn't have been able to migrate the existing projects. Along with the migration we set up **Microsoft Entra Active Directory groups** to allow centralized management of user groups.

Next up came the bigger challenge of **migrating the Jira issues**. Adapting our client's well-established processes from Jira to Azure DevOps was like translating poetry from one language to another. Each platform had its **own syntax and vocabulary**. This required not just technical proficiency but also a flair for innovation. We embraced the challenge, leveraging our technical expertise to find ingenious ways to maintain the integrity of their critical workflows in the new environment.

The Jira Jigsaw

Our client used Jira for issue tracking, and things were getting a bit tangled in there. Different projects, different fields – a **restructuring** was needed. That's where we came in, working in collaboration with the client to create a structured and streamlined approach to **issue tracking** using Azure Boards.

To achieve this, we employed an **open-source tool** for migration. When we discovered that the tool didn't support historical data, we didn't just lay our heads down – we fixed it! We **contributed with a pull request**, and now we're definitely proud that we were able to give back in some small way to the open-source community.

Additionally, we worked together with the client to determine the **ideal structure** for organizing items transferred via the Azure DevOps API. This restructuring transformed a chaotic list of ticket items into a **neatly organized system**, improving efficiency significantly.

Templates for Triumph

We didn't stop at migration; we also implemented **Azure Boards templates** to simplify the ticketing process within Azure DevOps. This not only saved time but also ensured consistency across our client's projects. Furthermore, we provided invaluable support for their new processes, adapting to their unique needs with ease.

In roughly six months, we migrated the client's infrastructure from Jira to Azure DevOps and **implemented Azure Boards templates** to simplify the ticketing process. Furthermore, we provided the necessary support for their new processes, adapting to their unique needs with ease.

– THE RESULTS

As we conclude our journey with our client, let's summarize the concrete business outcomes that emerged from our collaborative IT transformation.

Cost Reduction: Our client witnessed a significant reduction in IT costs, thanks to the optimization of licenses and resources.expiration.

GDPR Compliance: The migration to a European tenant ensured GDPR compliance, mitigating potential risks.

Streamlined Processes: CloudFuel's expertise led to a streamlined IT infrastructure, enhancing operational efficiency.

Improved Productivity: The adoption of Azure Boards templates simplified input and improved project management.

HEALTHCARE CLOUD MIGRATION SUPPORT

The healthcare sector today requires high levels of accuracy, speed, and safety, and hospitals like our client are adopting digital solutions to overcome these difficulties. Their large collection of applications, along with overstretched infrastructure teams, created an urgent demand for change. To solve this, they joined forces with CloudFuel, and we began a journey to move their on-site infrastructure to the cloud.

– THE CHALLENGE

Our client faced a **series of challenges** that demanded immediate attention. Overworked operations teams, burdened by the demands of an extensive portfolio of applications, were grappling with the complexities of maintaining outdated software. This resulted in a relentless cycle—outdated software required continuous efforts, exacerbating the strain on already overburdened teams.

In the realm of healthcare, where **data security** is sacrosanct, this situation presented a daunting dilemma. Our client was not merely wrestling with technology obsolescence but also confronting the need to safeguard sensitive healthcare data. Balancing the pressing need for modernisation with the importance of security was the challenge that CloudFuel was called upon to address.

Our collaboration was not just a migration project; it was a **strategic response** to these multifaceted challenges. Recognizing the critical importance of streamlined operations, modernisation, and security in the healthcare domain, CloudFuel embarked on a journey to revolutionize our client's IT landscape while ensuring that every step was taken with the utmost consideration for data security and privacy.

– OUR SOLUTION

At CloudFuel, we understood the unique challenges that our client was grappling with. To set the stage for this remarkable transformation, we began with a comprehensive **workshop**, aimed at forging a strategic path towards a digitally empowered healthcare ecosystem. Afterwards, we identified **three critical stages** that would bring their infrastructure up to speed.

Phase 1 Azure DevOps Server Migration

Our journey with the client began with a practical necessity: **upgrading** their existing infrastructure to the latest 2022 version of Azure DevOps Server, a crucial step before venturing into Azure DevOps Cloud. The challenge was significant, given that our client was still using the 2015 version.

Traditionally, transitioning from an older system to a newer one was a complex and time-consuming process. It required careful planning and a significant investment of time and effort. However, **Microsoft's migration wizard** proved to be a game-changer, simplifying the migration process significantly.

What would initially take us several weeks can now be accomplished in a manner of days. This wasn't revolutionary, but it was a welcome **efficiency boost**. This swift progress laid a sturdy technological foundation for our client's digital transformation journey, exceeding our and our client's expectations.

Phase 2

Transitioning to Azure DevOps Cloud

As our partnership advanced, we found ourselves in the midst of the second phase—migrating our client's operations to Azure DevOps Cloud. This transition promises to empower them with **enhanced scalability, flexibility, and improved collaboration capabilities**, positioning them to navigate the fast-paced healthcare landscape with agility and confidence.

Even as this phase is **currently underway**, we're already feeling quite confident about it. Why? Well, it all goes back to our experience in Phase 1. Reflecting on our successful migration, we couldn't help but feel a bit surprised by how smoothly it all went, but that sense of surprise has fueled our determination for Phase 2.

Our CloudFuel engineer has been diligently **testing the migration process** on a local setup, so that when the time comes for the full migration, it will be a well-practiced and seamless operation. This meticulous preparation underscores our commitment to making our client's journey towards enhanced scalability and flexibility a resounding success.

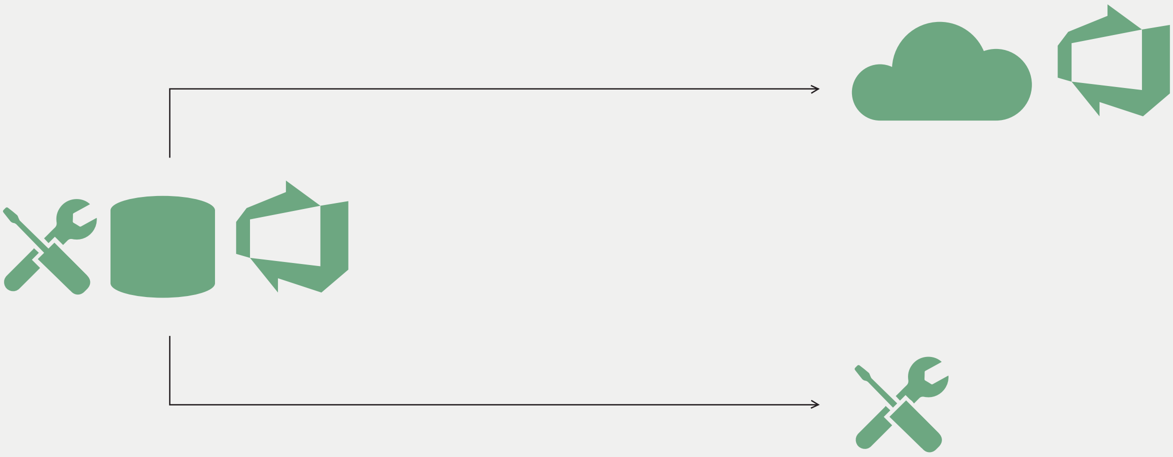
Phase 3

Merging Organizations and Elevating Security

Looking ahead, our final phase will involve the critical task of **merging organizations and strengthening security measures**. In an era where data security is paramount, we're committed to ensuring that no secrets are embedded within the code, fortifying our client's defenses against potential threats.

– GOING BEYOND MIGRATION

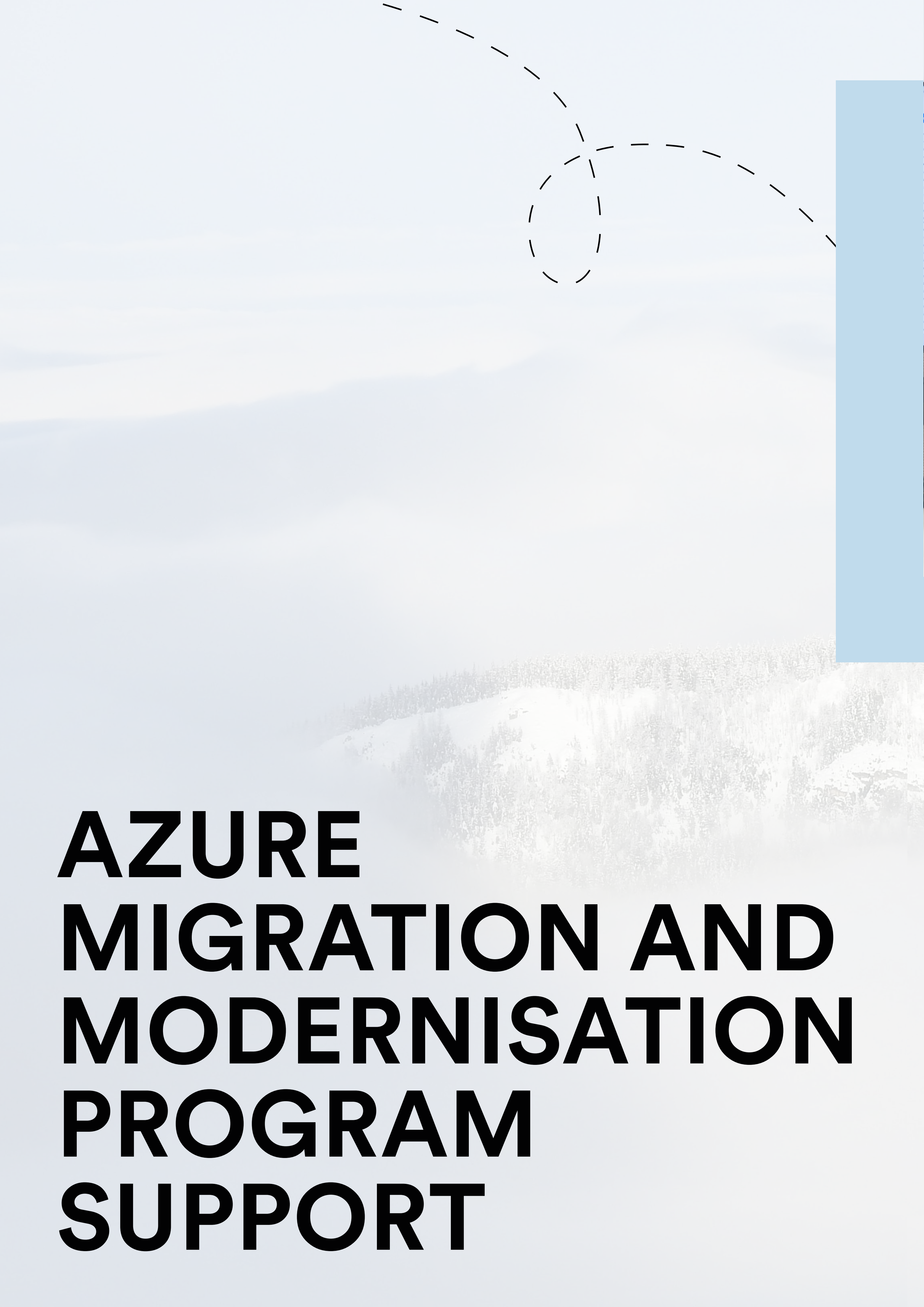
Beyond the core migration, CloudFuel's engagement with our client includes **additional enhancements** that will shape the hospital's digital future. We are facilitating the migration from Team Foundation Version Control (TFVC) to **Git**, a transition that streamlines code management and fosters collaboration. Additionally, we are orchestrating the shift from classic pipelines to **YAML pipelines**, further automating and optimizing our client's software development lifecycle.



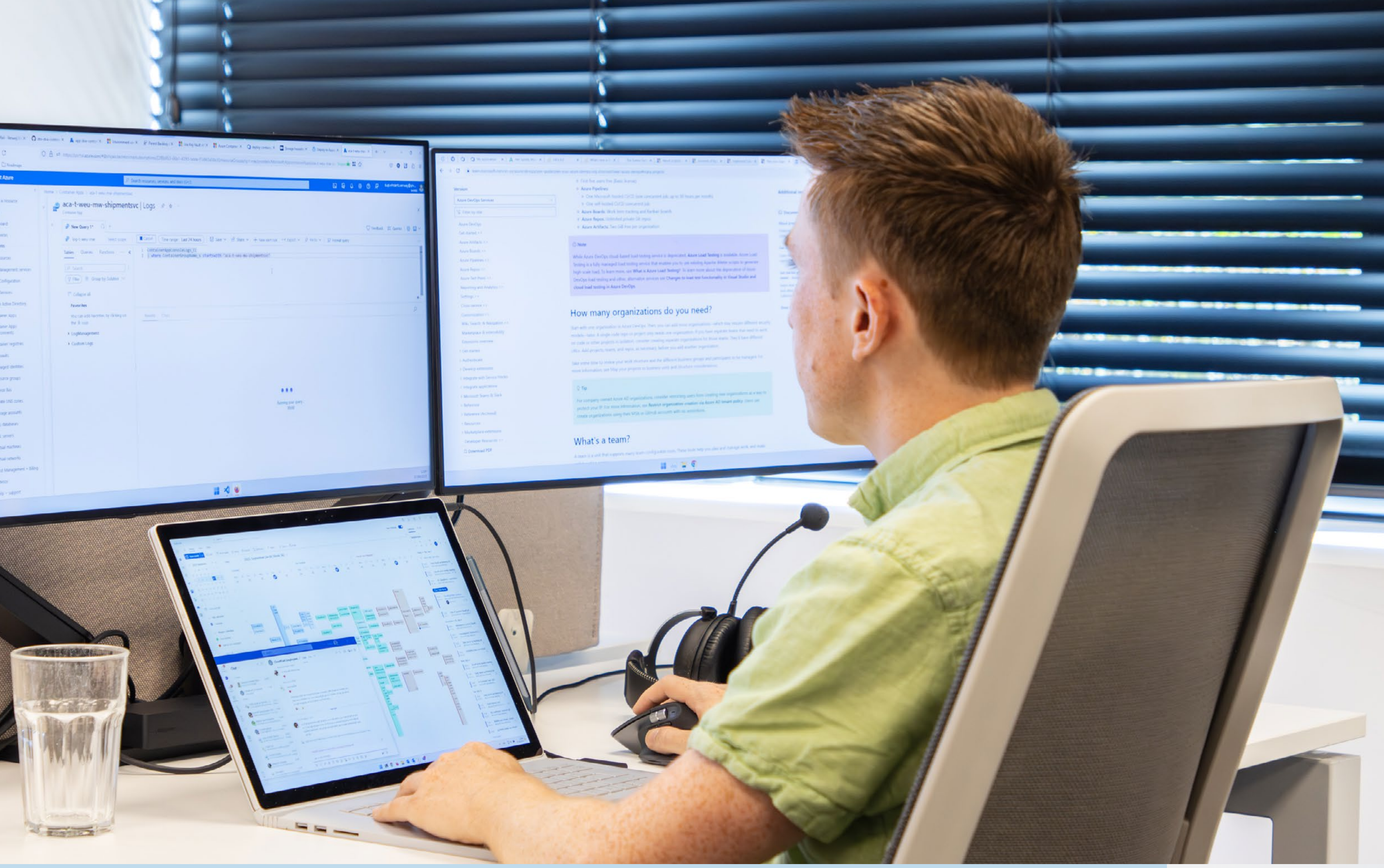
– CONCLUSION

The collaboration between our client and CloudFuel is not merely a migration project; it's a journey toward a more advanced, secure, and digitally enabled healthcare IT landscape. As they look forward to completing the transition to Azure DevOps Cloud and embracing state-of-the-art security practices, together, we are transforming the way healthcare IT operates.

In summary, this case study underscores how CloudFuel's technical expertise and strategic guidance are propelling our client toward a brighter, more digitally empowered future in healthcare IT. We are not just witnesses to this transformation; we are proud partners in shaping our client's digital success story.



AZURE MIGRATION AND MODERNISATION PROGRAM SUPPORT



In this case, we'll delve into the exciting project with our client: a major player in the water utilities sector. Their objective? To migrate their existing applications to Azure. Our journey together embraced insightful workshops, deep dives into Azure Container Apps (ACA) and Azure Kubernetes Service (AKS), and culminated into a tailored reference architecture, future-proofing their digital infrastructure. Join us as we unfold this case.

– THE CHALLENGE

Our client wanted to move their **existing microservices architecture to Azure**. Since they had some in-depth questions regarding compute solutions, DevOps, networking options, security solutions and more, our client realised that they could use an **expert's opinion**.

– THE SOLUTION

We helped our client on two separate occasions. First, we answered various questions related to Azure and replatforming in a series of **dedicated workshops**. In the second phase of our cooperation, we organised **technical deep dives** into their two preferred Azure services for running containerized applications: Azure Container Apps (ACA) and Azure Kubernetes Service (AKS). We then designed and refined a reference architecture together with our client to provide them a blueprint of their destination microservices architecture and a foundation for the next steps in their application modernization journey.

Phase 1 Discovering and Deciding

We commenced with an **intake meeting** where we delved into their requirements and examined their existing infrastructure. They operate **on an on-premises microservices** architecture rooted in **Netflix OSS**. The architecture supports a modest portfolio of under 100 applications and has multiple internal and external development teams working on it.

Once we got a good view of our client's current setup, we conducted a **series of workshops** that were tailored to address their specific inquiries and concerns. Using our expertise, we helped them with the following aspects of their transition:

Microservices and Java applications on Azure: in this workshop, we gave a high-level overview of various computing solutions focused on running containers in Azure, including Azure Spring Apps, Azure Container Apps (ACA), Azure Kubernetes Service (AKS), and Azure App Service.

Embracing DevOps in Azure: since DevOps is essential for a cloud-based ecosystem, we conducted a workshop centred around harnessing the power of streamlined development and deployment practices in Azure.

Hybrid environment setup: in another workshop, we focused on how our client could leverage the combined benefits of cloud and on-premises infrastructure. We highlighted the Azure VPN Gateway service, which uses Site-to-Site VPNs, as a recommended solution.

Service discovery in a hybrid scenario: our client wasn't sure if and how they could combine the service discovery of Netflix OSS and AKS. We recommended separating the discovery while ensuring that communication between their microservices would still be possible.

Setting up a landing zone: Arxus, our hosting partner, addressed our client's questions on establishing an effective landing zone within the Azure environment in another workshop.

Armed with new insights and information on various aspects of architecture migration, our client took some time to conduct **further research** internally. After a few months, they approached us again to help them with the next step in their application modernization journey.

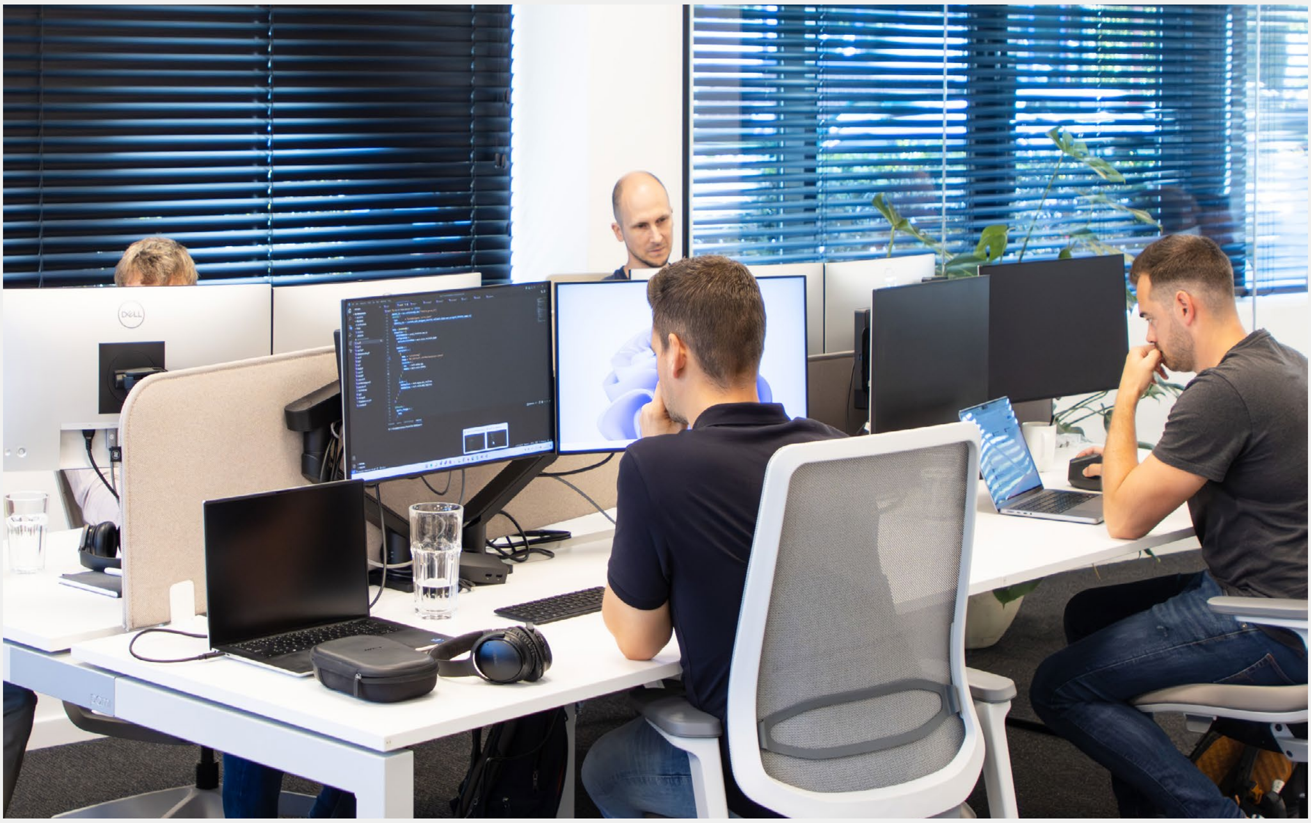
Phase 2 Deep Dives and Design

After this period of internal research, we started up a second project that focused on the **technical details** of their future(proof) architecture, instead of the previous phase's high-level overview.

Our client had narrowed down the options for a computing solution to Azure Container Apps (ACA) and Azure Kubernetes Service (AKS), but needed some additional guidance to make the final decision. We facilitated **deep dives** for both options, carefully weighing the pros and cons of each solution based on their unique requirements. For example, our client wanted a Kubernetes cluster that allowed for separate rights management for each of their development teams, both internal and external, to guarantee the security of their infrastructure.

Once our client made their decision, we started work on drafting a **reference architecture** that took their needs into account. Then, we organised a thorough **review** of the draft, ensuring that every facet aligned with their vision of the destination infrastructure. Drawing upon their valuable insights and feedback, we refined and honed our draft, culminating in a robust and comprehensive final reference architecture.

> Via the workshops, **we provided guidance** on resources, a branching strategy, and a roadmap for all stakeholders. Next, we advised on which storage solution would work best for the large files used in the application.



– CONCLUSION

Our client is well-prepared to move forward with their application modernisation journey, thanks to their refined reference architecture and in-depth Azure insights. We remain at their disposal for guidance, assistance, or expertise during the implementation and configuration phases.

DEVELOPMENT COACHING FOR A CLOUD-NATIVE APPLICATION

Sometimes, developing an intricate application requires not just external help with coding, but the perspective of an expert. In this case study, we'll explain how our resident Azure Cloud Native Architect coached both the internal and external developers at one of our clients in the creation of a standalone cloud-first application.

– THE CHALLENGE

Our client offers an all-in-one **asset management platform** that focuses on product packaging. It boasts several user-friendliness and collaboration features, including a powerful **viewer**. This viewer allows users to inspect and modify assets through a layered 3D model view and offers colour separation, which is essential for printing.

Despite its strong reputation, the viewer was relatively slow to use. To further improve the user experience, the client's team decided to create a **standalone, cloud-native viewer application** with the help of an **external partner**. They contacted us through a mutual connection and asked us to guide both development teams with our expertise in architecture, application modernisation, and cloud-native development.

– WORKSHOPS

We started with **two engaging workshops**, designed to ignite enthusiasm among the internal development team for the Azure platform and familiarise them with its capabilities.

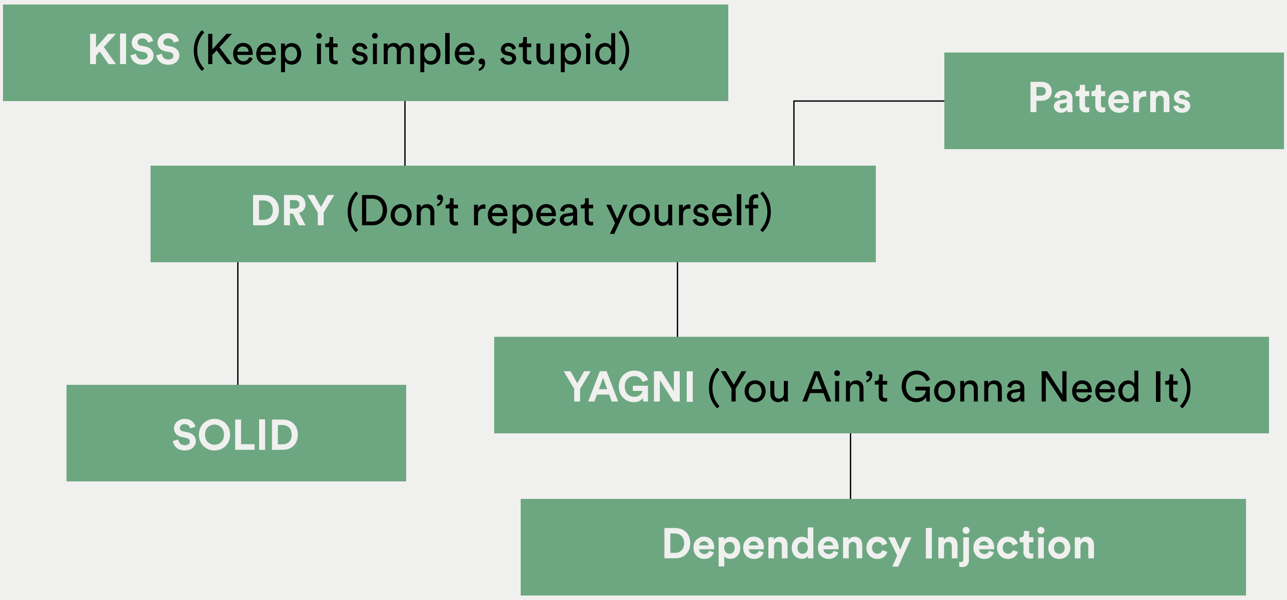
In the first workshop, we compared **Azure computing solutions** based on our experience in application modernisation: Azure App Service, Azure Container Apps, Azure Kubernetes Service, and Azure Functions. We recommended the team to start with App Services and later transition to ACA or AKS, which they agreed to.

The second workshop was a comprehensive two-day session on **automatic deployment** and DevOps through CI/CD pipelines. Using a PHP-based proof of concept (POC) similar to our client's application, we demonstrated manual deployment to Azure, introduced Terraform for automation, and explored deployment options through YAML and Classic pipelines.

– ANALYSIS AND CODE REVIEW

With the workshops completed, we conducted a full-day **analysis session** with our client's CTO. We examined the viewer's features, identified bottlenecks, and provided some recommendations. These included the compute solution choice mentioned earlier, and how to best integrate Datalogics into their upcoming application.

We also facilitated a **code review** of the external party's .NET work once our client had called for a temporary hold on development. We offered some improvements in areas such as consistent dependency injections, variable declarations, and avoiding so-called magic strings. We didn't spot any serious errors, but we were able to suggest some improvements in **development best practices**.


KISS (Keep it simple, stupid)

Patterns

DRY (Don't repeat yourself)

SOLID

YAGNI (You Ain't Gonna Need It)

Dependency Injection

– DEVELOPMENT COACHING

As the external party resumed development, we decided to take a step back and examine how they should best proceed. We provided their developers with guidance on **resources, a branching strategy, and a roadmap** for all stakeholders. Our advisory role extended to setting up **CI/CD pipelines** correctly through Terraform, building on what the internal development teams learned in our second workshop. Finally, we examined which **storage solution** would work best for the relatively large files used in the viewer.

At the moment of writing, we're **still guiding** the development process. For example, we recently helped with spikes in **memory usage**. Our client used the Azure consumption only program, which has a 4GB limit of working memory. We recommended using the **workload profiles** functionality instead, which has dedicated plans with higher limits. We also pointed out the difference between general-purpose and memory-optimised workload profiles for DEV and PROD environments.

Kristoff also completed a **code review of the backend**. This revealed some errors that lead to files being downloaded after every change, including zoom-ins. Of course, there were significant **performance gains** to be found here. Frontend-wise, we also advised to **optimise calls** to the backend based on user searches, and a debounce time.

– LOOKING AHEAD

As the project continues, we join the client's CTO for **strategic discussions** at regular intervals to discuss various aspects of the application. For example, they recently asked us to test which framework would be best suited for the colour picking functionality of the new viewer. We assisted them with a comparative benchmark.

In the future, we will also help our client and its architect in shaping the **architecture** for the viewer application. As the external development party progresses toward delivering the minimum viable product (MVP) and the first customers are onboarded, our focus will shift to considerations like **security** and the **landing zone**. We're excited to see how the new viewer project will progress.

– CONCLUSION

In conclusion, we appreciate your interest in CloudFuel's whitepaper on cloud migration and modernization. The provided case studies exemplify our proficiency in delivering customized solutions to assist clients in navigating the intricacies of these processes.

The success stories underscore CloudFuel's ability to offer innovative solutions and a thorough approach, resulting in effective deployments, cost-effective scaling, and heightened security for our clients.

If you are currently seeking a partner for your cloud migration and modernization needs, we encourage you to consider CloudFuel. Our proven track record and commitment to excellence position us as a reliable choice for addressing the complexities of this transformative journey.

Feel free to contact us to learn more about how CloudFuel can assist your business in achieving its cloud migration and modernization objectives. Thank you for downloading our whitepaper, and we welcome any inquiries you may have.

ARE YOU LOOKING FOR AN EXPERT PERSPECTIVE TO GUIDE YOUR LATEST DEVELOPMENT PROJECT?

At CloudFuel, we specialise in guiding organisations in their application modernisation journeys to the cloud.

Contact us and we'll gladly discuss the possibilities!

Info@cloudfuel.be
Veldkant 33A
2550 Kontich
+32 498 88 15 74

Cloud Fuel