Microsoft Azure

How to modernise your apps: unlock the potential of AI with Azure

Contents

WHY MODERNISE YOUR APPS?

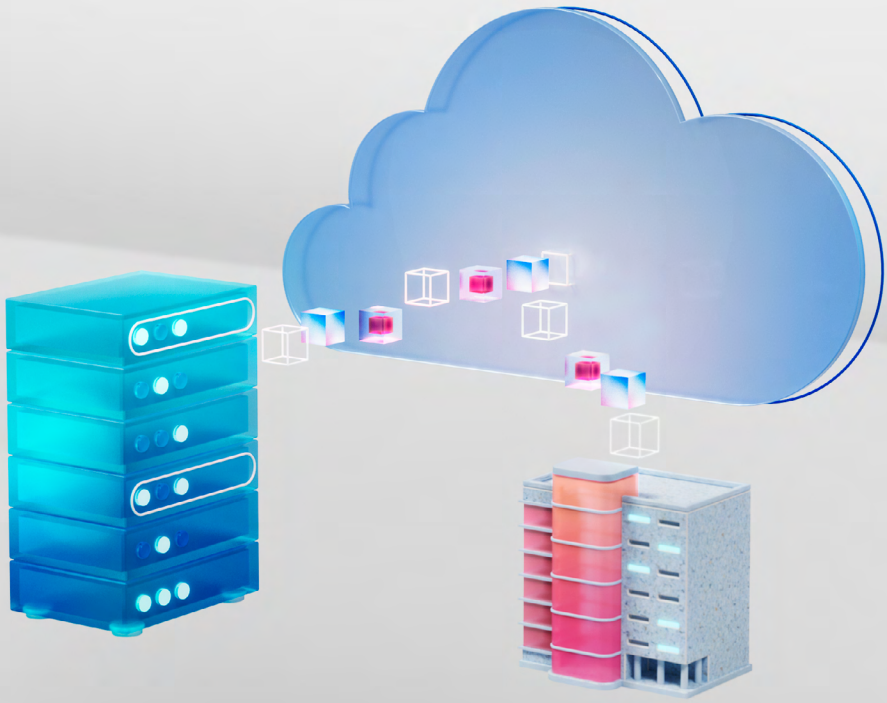
Building the next generation of apps

Application modernisation is the process of updating legacy apps, data and infrastructure to a cloud-first model, to improve performance, drive innovation and integrate new technologies like generative AI.

The growth of cloud computing and generative AI is making it possible for developers to create apps with exciting new capabilities. Intelligent apps incorporate AI and machine learning, meaning they can offer better user experiences through personalised features, automated processes and conversational interfaces. They can incorporate new functions quickly and respond flexibly to real-time data, learning and adapting to meet user needs.

Cloud platforms such as Microsoft Azure offer the modern infrastructure and services needed to develop, deploy and manage these applications. In turn, AI-infused apps are driving digital transformation, allowing businesses to innovate rapidly while optimising efficiency and maintaining robust security.

This e-book explains the key architectures, processes and tools you'll need to start modernising your apps, data and IT infrastructure to make the most of these opportunities. You'll find useful links to explore these topics further, and real-world use cases highlighting the impact intelligent apps are already making.

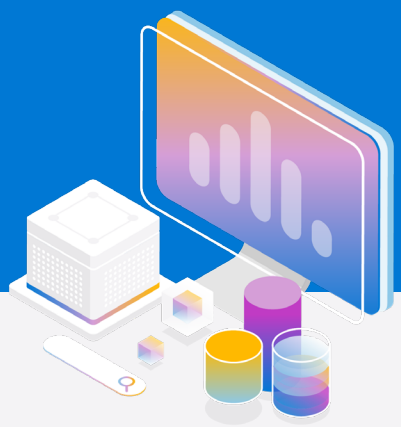


New technology for new challenges

As technology changes, so do the challenges and opportunities facing businesses. Companies know they must innovate to meet growing consumer expectations in an unpredictable economic environment. And as developers build systems and services to meet new requirements, they face considerable cost and time pressures alongside ever-evolving security risks.

With their combination of cloud technology and AI, intelligent apps have the agility and power to meet these demands. Cloud systems support greater cost efficiency and dynamic scalability as well as faster, more iterative development processes. Meanwhile, generative AI offers the capacity for automation, personalisation and in-depth data analysis.

This means that with the right cloud platform and tools, developers can rapidly build, launch and maintain apps that unlock new value for their organisation and its customers.



Intelligent apps help organisations boost productivity, innovation and efficiency in a variety of ways, including the ability to:

Build custom copilots to provide personalised and instant assistance to employees and customers.

Automate repetitive tasks to free up time for creative and strategic work.

Accelerate decision-making processes using real-time data and analytics.

Gather deeper insights into customer needs and preferences by analysing user feedback and behaviour to deliver more personalised experiences that improve over time.

Support engaging and natural interactions between people and software using generative AI.

What can you do with AI apps?

What you plan to do with an app will affect the way you create it. So it's important to think about this early on, considering your organisation's most pressing needs as well as new opportunities. For example, it could benefit a business to:

Create differentiated products
Build innovative products and services with AI capabilities to grow your business and tap into new markets.

Accelerate content delivery
Create personalised marketing content across platforms, from social media posts to product descriptions.

Transform customer service
Provide personalised and interactive responses to answer customer questions and facilitate routine tasks.

Personalise customer experiences
Deliver tailored content, products and services to specific users based on their behaviour, preferences and needs.

Optimise employee workflows and processes
Empower people with rich knowledge when they need it and automate manual tasks to free up time for higher value work.

Prevent fraud and detect anomalies
Handle high volumes of transactions reliably, and identify patterns, anomalies or suspicious activity.

Unlock organisational knowledge
Surface insights from vast amounts of data and make them accessible through natural language interactions.

Automate document processing
Classify, extract, summarise and gain deeper insights from documents using natural language prompts.

Intelligent apps in action: real-world examples

BMW Group uses Azure-powered IoT for 10 times more efficient data delivery.

Read BMW Group customer story >

> **"Azure is the turbocharger for delivering the right data to the right person on a large scale."**
>
> Christof Gebhart, Mobile Data Recorder Co-Creator, BMW Group

JATO Dynamics saves customers 32 hours a month with automated, tailored content.

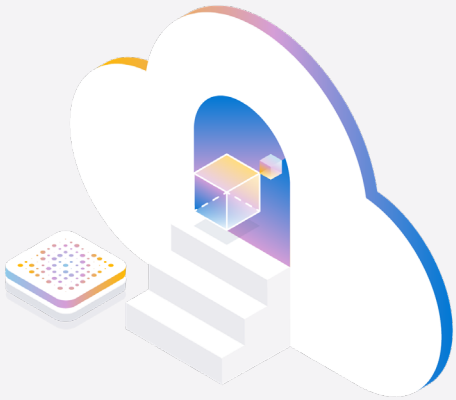
Read JATO Dynamics customer story >

Workflow automation drives 58% higher job retention at **Medigold Health**

Read Medigold Health customer story >

CREATING AN AI-READY ARCHITECTURE

Architecture patterns for intelligent apps



Traditional network architecture uses physical data centres to store digital assets and operate network systems for daily functions. In this set-up, a user's access to data, software or storage is restricted to the device or official network they're connected to.

In contrast, cloud architecture provides on-demand access to shared computing resources such as servers, storage, apps and services via the internet. It lets you scale resources as needed and effectively handle dynamic workloads without running into failures.

Because traditional apps are hard-coded with fixed data sets, they offer constrained interactions and are costly and complex to change. Meanwhile, cloud infrastructure supports intelligent apps to provide data-driven, personalised experiences through conversational interfaces, and continuously improve through learning.

There are different ways of building intelligent apps to take advantage of cloud computing capacities. Let's look at some of the main architecture options.

Microservices

Cloud-based apps typically use a microservices architecture – a collection of small, autonomous services that are self-contained and provide a single business capability. Unlike traditional monolithic architectures, microservice architectures are scalable – meaning you can scale the individual components of an application up or down independently. They're also more resilient, because the rest of the application can function normally even if one element fails.

Because the microservices are isolated from each other, developers can make changes to one component without impacting the rest of the application. This means you can update an existing service without rebuilding and redeploying the entire app.

Explore microservice architectures >

Event-driven architecture

With event-driven architecture (EDA), an app is built as a collection of decoupled services that can communicate with each other by delivering events between producers and consumers. This makes them highly scalable and responsive.

Typically implemented through microservices, EDAs operate in real-time. They respond to "events" (changes in state) ranging from user interactions to instantaneous analytics. An event could be a website visitor putting an item in an online shopping basket, a sensor reading from an industrial site or an inventory update on warehouse stock, for instance.

Explore event-driven architectures >

Containerisation

Moving apps from one environment to another can cause them to malfunction. These issues often stem from disparities in configurations, underlying libraries and dependencies. Containerisation helps avoid malfunctioning by bundling an app or service with its dependencies and configuration into a lightweight, portable container image.

This containerised app can then be deployed on the host operating system as a cohesive unit. Containers enhance efficiency by helping you deploy applications consistently across different computing environments.

Learn about container orchestration for microservices >

There are various further design options for intelligent apps, each requiring different approaches and technology. For example, **serverless architecture** allows a cloud provider to dynamically manage the app infrastructure, so developers can focus on writing code. And **API-first design** prioritises the development of application programming interfaces (APIs) before implementing other components.

Figure 1 compares modern architecture patterns and their typical transformation approaches. To learn more about different approaches, visit the **Azure Architecture Center**.

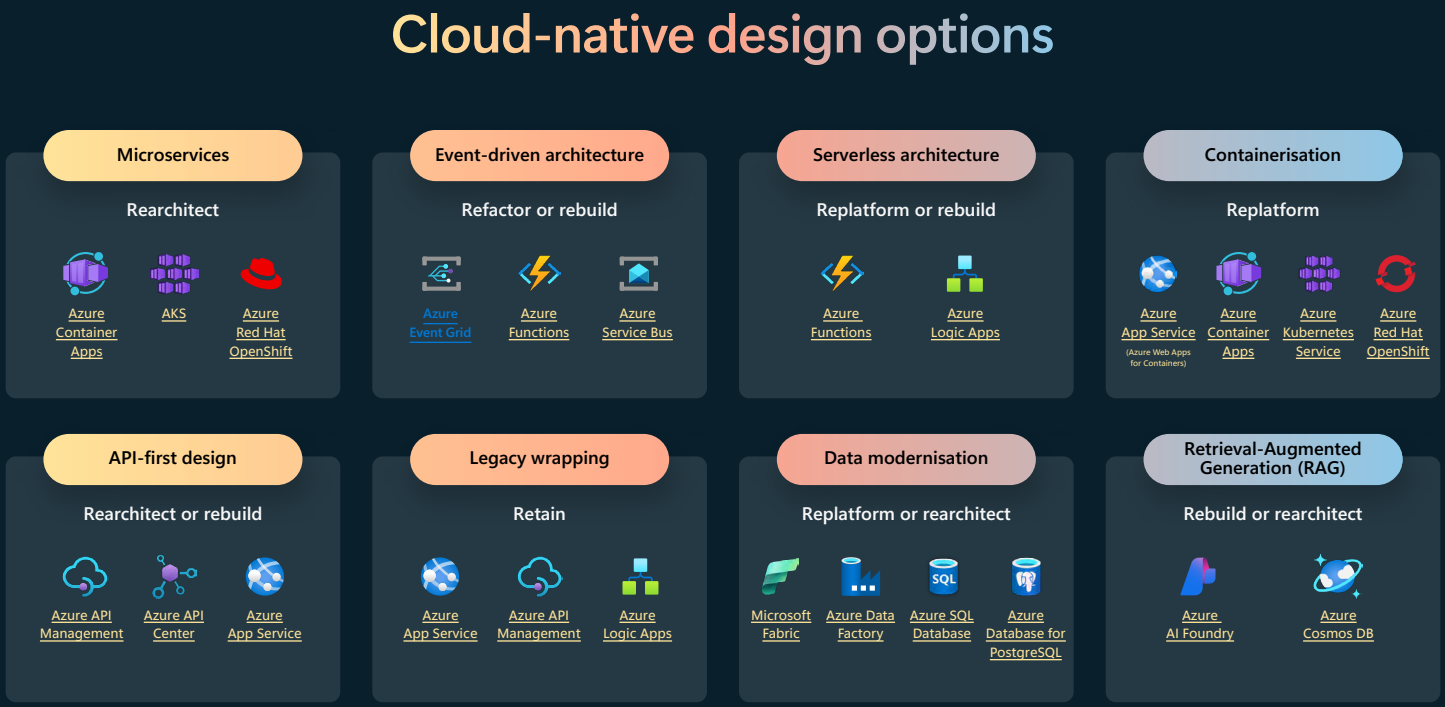


Figure 1. **Architecture patterns and transformation strategies.** This diagram maps architecture types such as microservices, event-driven, and serverless to recommended approaches like rearchitecting, replatforming or retaining, with corresponding Azure technologies.

Paths to modernisation

Alongside considering the architecture you want to use for an intelligent app, it's important to think about how you'll get there.

Modernisation can involve a wide range of activity – from migrating apps to the cloud largely unchanged, to substantially redesigning their architecture or building completely new capabilities.

The best strategy will depend on each individual situation – and will often involve a combination of approaches that evolve over time.

Learn more about app modernisation >

> **"Azure PaaS allows faster deployment of environments and services that can be scaled up to meet demand—we're using elasticity in a much better way"**
>
> Daniel Engberg, Head of AI, Data, and Platforms, SAS - Scandinavian Airlines

Read SAS - Scandinavian Airlines customer story >

Modernisation strategies

Depending on the nature of your current apps and your business goals, there are a range of modernisation options to consider.

Rehost

Migrating an app to a virtual machine in the cloud ('lift and shift').

Replatform

Moving an app to a cloud runtime platform with minimal code changes ('lift, tinker and shift').

Refactor

Restructuring complex code into microservices, optimising an app for the cloud without changing its external behaviour.

Rebuild

Building completely new capabilities for an app using a cloud-native architecture pattern.

Retire

Decommissioning or shutting down existing apps.

Retain

Continuing to use legacy apps for the time being.

On-demand videos

Watch "Innovate, deploy, & optimize your apps without infrastructure hassles" > (59 minutes)
Watch "From legacy to cloud native: Scaling Modernization with Azure and AI" > (59 minutes)
Watch "The future of .NET app modernization streamlined with AI" > (61 minutes)
Watch "Build and scale your AI apps with Kubernetes and Azure Arc" > (43 minutes)
Watch "Use agentic AI to simplify .NET upgrades with GitHub Copilot" > (15 minutes)

HOW AZURE CAN SUPPORT YOUR MODERNISATION JOURNEY

The modernisation lifecycle: step-by-step

Modernisation isn't a one-off event – it's a continuous process of improvement and optimisation. It requires a systematic approach to make sure you're constantly assessing and adapting your activity to meet evolving needs and opportunities.

Microsoft Azure supports organisations to modernise through a structured process involving four key stages: assessing the current state of your apps and data, planning the modernisation, executing your strategy, and then maintaining and optimising the new technology.

These stages, along with their key related activities, are shown in Figure 2.

Learn more about the app modernisation lifecycle >

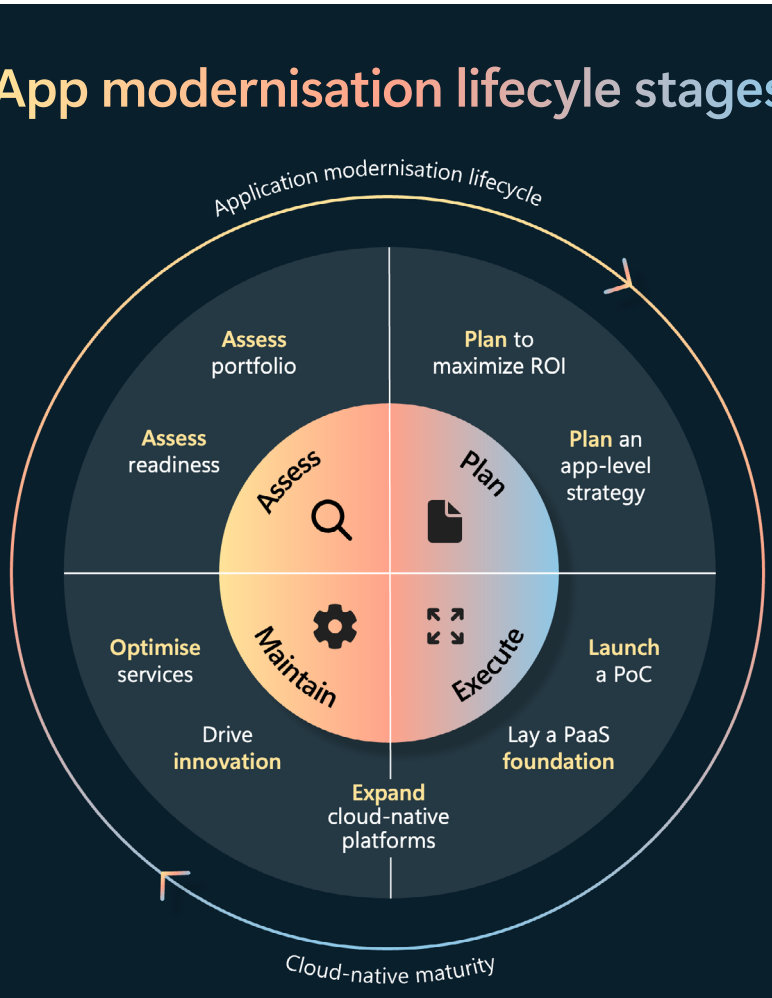


Figure 2. **App modernisation lifecycle stages.** A circular diagram showing the lifecycle phases of Assess, Plan, Execute and Maintain. The quadrants include key activities, from assessing readiness and planning for ROI to launching a proof of concept and optimising services.

1. Assess

To understand where to start your journey, the first step is to carry out a careful assessment of your existing apps, data and IT infrastructure. You should look for opportunities to modernise while evaluating costs and organisational readiness. This involves:

- Identifying legacy components that need upgrading or replacing.
- Evaluating architecture to determine how ready it is for cloud deployment.
- Analysing performance bottlenecks and security vulnerabilities.
- Assessing the risks, costs and benefits of modernisation.

Learn more about modernisation assessment >

Microsoft Solution Assessments

To make sure they prepare thoroughly for modernisation, many businesses choose to work with one of our official partners through a Microsoft Solution Assessment. This expert guidance can help you minimise risk, save time, reduce costs and gain support for cloud migration across the organisation. You can also benefit from funding support, which in many cases covers the full cost of the assessment.

Learn about Microsoft Solution Assessments >

2. Plan

Once you've completed the assessment, it's time to create a detailed modernisation plan. This involves deciding which apps to modernise, which to replace, and which to build from scratch. You can use the six Rs framework to help guide your decision-making and consider the amount of work required to meet your objectives. Effective planning ensures resources are allocated efficiently, stakeholders are aligned and goals are clearly defined.

The modernisation plan should act as a roadmap outlining:

- Priorities for updating apps, data and workloads.
- The target architecture on Azure.
- The services and resources needed.
- A timeline, budget and milestones for the migration process.
- Contingency strategies.

Learn more about modernisation planning >

The Azure Migrate tool can help you assess apps to identify key code changes required before migrating to Azure, and build a modernisation plan.

Application and code assessment tool >
Build a migration plan with Azure Migrate >

Air India's virtual assistant

Air India used Azure tools to create a generative AI-based virtual assistant, which now handles 97% of queries with full automation, saving millions of dollars in customer support costs.

Read Air India customer story >

> **"For every business requirement, we were able to figure out how we could realise it on the Azure platform. The support from Microsoft has been amazing, both in terms of technical support and co-innovating solutions together."**
>
> Viju Chacko, Head of Digital Architecture, Air India

3. Execute

This is the moment of truth. Execution involves using modern frameworks, cloud-native architectures, and AI capabilities to transform your applications and data. Effective deployment ensures these systems are rolled out seamlessly with minimal disruption, laying a strong foundation for a cloud-first approach.

Launch your application modernisation strategy >

Azure is a secure, reliable platform that supports modernisation at every stage. Including developer services, application services, databases and AI services, it offers a wealth of resources and expert support to ensure a seamless process – wherever you are on your journey.

With Azure, you'll gain access to a wide-ranging suite of support including:

- Comprehensive, AI-ready platform services
- Enterprise-grade, end-to-end security
- Engineering support to follow a proven methodology
- Support for all languages and frameworks
- Enterprise skilling and enablement
- An ecosystem of specialist partners

Figure 3 schematically sets out the modernisation stages along with Microsoft support.

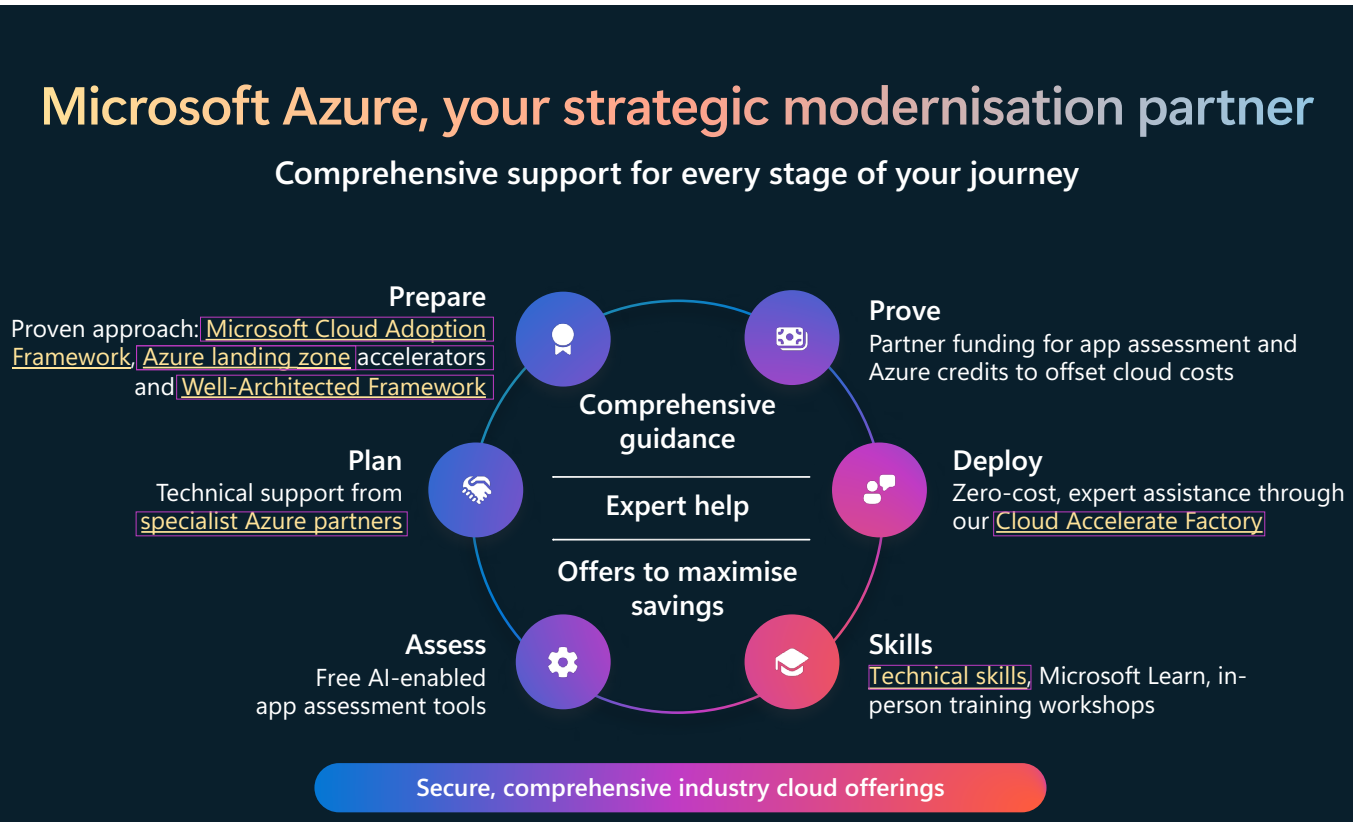


Figure 3. **Supporting the Azure modernisation journey.** A circular diagram showing six modernisation stages, from assessment to deployment and skilling, and the support available at each step.

Specific Azure services that support modernisation include:

Azure App Service
Enables developers to build and host web apps, mobile backends, and RESTful APIs.

Learn more about Azure App Service >

Azure Functions
Provides serverless computing capabilities, allowing you to run code in response to events without managing infrastructure.

Learn more about Azure Functions >

Azure Kubernetes Service (AKS)
Supports the open-source deployment, management, and scaling of containerised apps using Kubernetes.

Learn more about Azure Kubernetes Service >

Azure Container Apps
Lets you deploy apps and microservices from code or containers without the need for intricate infrastructure orchestration.

Learn more about Azure Container Apps >

Azure SQL Database
Offers managed relational database services with built-in intelligence and security features.

Learn more about Azure SQL Database >

Azure Cosmos DB
Provides a globally distributed, multi-model database service for scalable and high-performance apps.

Learn more about Azure Cosmos DB >

GitHub
Brings together development, testing, and deployment processes to ensure continuous integration and continuous deployment (CI/CD)

Learn more about GitHub >

Figure 4 shows the full Azure cloud app platform stack, including AI services, integration tools and observability features.

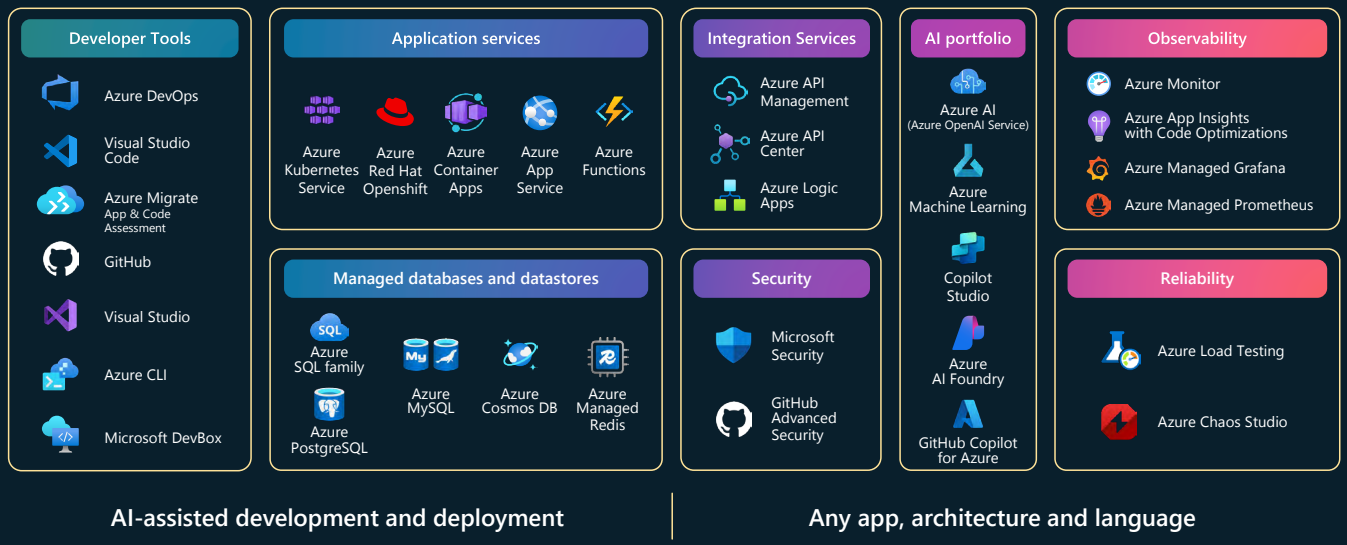


Figure 4. **Azure cloud app platform capabilities.** This chart shows Azure's platform components for AI readiness, grouped into developer tools, application services, integration services, AI portfolio, observability, reliability, databases and security.

4. Maintain

Modernisation is an ongoing process. Once you've moved your mission-critical apps to the cloud, you're ready to focus on expanding capabilities, driving innovation, and optimising performance. This could involve:

- Adopting DevOps and DevSecOps practices to streamline workflows and accelerate delivery.
- Monitoring app performance using Azure Monitor for full-stack visibility, including Azure Monitor Application Insights for detailed metrics and proactive anomaly detection.
- Implementing security best practices, including the use of Microsoft Defender for Cloud and Microsoft Entra ID for identity management.
- Optimising costs by using Microsoft Cost Management to track usage and Azure Reservations for long-term discounts.
- Using Microsoft Copilot in Azure for free, personalised recommendations that help you improve scalability, security, reliability and cost-efficiency post-migration.
- Ensuring compliance with corporate policies and regulatory requirements with Azure Policy.

Transitioning to the cloud is always challenging, with many factors playing a part in success. As well as working with a trusted partner, it's important to deploy the right approaches for your situation.

In the next pages we'll explore how AI-powered coding and security tools can support a successful modernisation journey.

> **"Because we use Azure services, Kinectify can provide agile delivery, rapidly deploying changes, enhancements, and bug fixes to our environment at a rate that on-premises solutions cannot match."**
>
> **Michael Calvin**, Chief Technical Officer, Kinectify

Kinectify: tackling financial crime in gaming

Kinectify used Azure services to build an anti-money laundering platform for the gaming industry that integrates data into a single view and workflow – allowing it to detect 43% more suspicious activities and achieve 96% faster decisioning.

Read Kinectify customer story >

Accelerate modernisation with GitHub Copilot

AI can empower developers to get more done – accelerating routine tasks and leaving more time to focus on innovation.

GitHub Copilot, integrated with the Azure app platform, is an AI pair programmer that can significantly boost productivity during modernisation. As well as recommending specific script, it can interact conversationally to explain how existing code works and guide your overall approach to projects. Here are some of the ways it can help:

Code suggestions GitHub Copilot provides real-time code suggestions based on context, reducing the amount of boilerplate code and speeding up development.

Learning and adaptation The tool learns from your coding patterns and adapts its suggestions accordingly – meaning it makes more accurate and relevant code recommendations over time.

Integrations GitHub Copilot integrates seamlessly with popular development environments like Visual Studio Code, making it easy to use during the entire modernisation process.

Generating documentation It can help with generating comments and documentation, ensuring that code is well-documented and making it easier to maintain.

Learn more about GitHub Copilot >

Capita uses GitHub Copilot to free up devs and drive customer service

By extending GitHub Copilot adoption across its software development team, Capita is uplifting the experience of both its developers and its customers.

Read Capita customer story >

> **"Developer productivity is about more than speed. Our developers feel more fulfilled and less frustrated when coding and that's a big win."**
>
> Michael Noonan, Managing Director of Software Solutions at Capita

What can you use GitHub Copilot for?

Refactoring code

GitHub Copilot helps you restructure code without changing its behaviour – whether that's to improve readability, reduce complexity, or make it easier to add new features, for instance. It supports developers with tasks including:

- Understanding the purpose of code and how it works
- Optimising inefficiencies
- Cleaning up repetition
- Making code more concise

Learn more about refactoring code with GitHub Copilot >

Migrating a project to another programming language

Moving a project to a different language can be difficult and time-consuming. GitHub Copilot can support this process by explaining the changes you need to make and suggesting replacement code in the new language. For example, it can:

- Outline the main steps you need to take
- Provide detailed guidance on each stage
- Help you fix bugs in the new code
- Support you with refactoring the code in the new language

Learn more about migrating to a new language with GitHub Copilot >

Writing tests

You can use GitHub Copilot to help you carry out unit tests to verify blocks of code, and integration tests to ensure the various parts of the system work together correctly. Through straightforward prompts, the tool can:

- Generate tests to cover a range of scenarios
- Validate performance for edge cases and exceptions
- Add further test cases
- Suggest tests you've overlooked to improve functionality

Learn more about writing tests with GitHub Copilot >

Modernising legacy code

Code that is old or no longer supported can be difficult to maintain or extend, as well as containing security vulnerabilities. GitHub Copilot helps you modernise legacy code by suggesting refactors and creating tests to catch potential issues. The tool can:

- Explain and provide diagrams about how legacy code works
- Provide suggestions for refactoring to follow modern best practices
- Generate tests to help make sure your changes don't introduce bugs
- Work with you iteratively to improve and refine functionality

Learn more about modernising legacy code with GitHub Copilot >

> **"Time saved with GitHub Copilot varies between 20% to 30%, and in some cases going as high as 50%."**
>
> Theo Mabaso, Group Chief Information Officer, Sanlam Life Insurance

Read Sanlam Life Insurance customer story >

Step-by-step – modernising COBOL with GitHub Copilot

To help understand how GitHub Copilot can help, let's look at a practical example of how you can use it to help modernise COBOL, a legacy language that developers often encounter. This step-by-step guide outlines a methodical approach to translating COBOL files into Node.js – ensuring a seamless switchover while maintaining the original code's integrity and functionality.

Explore how GitHub Copilot can help boost your productivity >

1 Compile and run the program

To compile the legacy COBOL program, you can either install a COBOL compiler, or use Copilot Chat in GitHub Codespaces. Either way, GitHub Copilot can carry out the task from a simple command.

2 Explain the files and code

Before you can modernise legacy code, you need to understand how it works. GitHub Copilot can provide a high-level overview, as well as detail on the logic of each file and the links between them.

3 Chart out the data flow

To help you further understand how the legacy program works, ask GitHub Copilot to produce a data flow diagram for the app in Mermaid to illustrate how the files link together.

4 Generate a test plan

Prompt GitHub Copilot to create a plan for tests to validate how the new app works as expected in all the necessary scenarios, and produce this in a markdown file.

5 Convert and review

It's now time to convert each file into the new language using GitHub Copilot. After carefully reviewing the output code, prompt the tool to link them together into a Node.js project.

6 Generate and run tests

Finally, ask GitHub Copilot to create unit and integration tests to ensure the accuracy and reliability of the new Node.js code. If a test fails, Copilot can also help you debug and refine the app.

Learn more about modernising legacy code with GitHub Copilot >

Streamlined and robust

Translating legacy apps into modern code is a complex process that requires careful planning and execution. By using AI tools like GitHub Copilot, developers can streamline their approach, ensuring that the new system is robust, reliable, and aligned with business needs.

Make the most of GitHub Copilot features >

Iceland: Accelerating innovation with GitHub Copilot

As it creates AI-powered apps to help it "do business at the speed of thought", Iceland's technology team is using GitHub Copilot to accelerate development – for example, by supporting developers to work with different languages.

Read Iceland customer story >

> "The power of GitHub Copilot is that it can do so many different things. It's really helped out in getting more unit tests into our apps – there's no excuse not to include them now."
>
> Craig Robinson, Software Engineering Manager, Iceland

Securing your code

As apps become a primary target for cyber threats, their security has a growing impact on businesses' finances, reputation and operations. Providing solutions in minutes rather than months, GitHub Advanced Security supports developers and security teams to work together to eliminate security debt and prevent new vulnerabilities. It's available through two standalone GitHub products: Secret Protection and Code Security.

Secret Protection

Unintentional leaks of secrets – such as such as API keys, passwords, tokens, encryption keys, and other credentials – during app development is a growing problem. GitHub Secret Protection is a tool that detects and prevents these secrets being exposed. It integrates seamlessly into workflows with minimal impact on efficiency.

Detection and prevention of secrets

GitHub Secret Protection combines proactive measures, such as blocking secrets before they are committed, with AI-powered detection capabilities. It can identify over 300 token types, and unstructured secrets like passwords, across code repositories, pull requests, and more.

Effective remediation support

By scanning entire Git histories and massive codebases across languages and formats, the tool helps ensure comprehensive visibility of leaks. Partnerships with 150 service providers, as well as low false-positive rates, support the efficient management of credentials to maintain productivity.

Find out more about GitHub Secret Protection >

Code Security

GitHub Code Security is a comprehensive suite of tools that enhance app security by integrating seamlessly into workflows. It allows you to identify and remediate vulnerabilities early in the software development lifecycle without holding back productivity.

Developer-first integration

GitHub Code Security is embedded within the GitHub platform. This allows developers to address security issues in their existing workflows, through tools like CodeQL for static analysis and Dependabot for dependency scanning.

Enhanced alert management

The solution reduces false positives and alert fatigue through precise analysis, improving developers' trust and increasing response rates.

AI-assisted remediation

Features like Copilot Autofix and Security Campaigns support quick and scalable fixes across repositories, streamlining the process of managing vulnerabilities.

Shift-left security approach

By integrating security measures at an early stage, GitHub Code Security allows developers to identify and resolve issues as they write and review code.

Find out more about GitHub Code Security >

SUMMARY AND FURTHER RESOURCES

Key takeaways

- By combining cloud technology with AI capabilities, a new generation of apps is helping organisations boost productivity, optimise operations and drive innovation.
- Modernisation involves updating legacy apps and data to a cloud-first, AI-ready model to take advantage of these opportunities.
- It can use a range of architecture patterns, including microservices, event-driven architecture and containerisation.
- Modernisation is not a one-off event, but an ongoing process involving four stages: assess, plan, execute and maintain.
- Microsoft Azure acts as a strategic modernisation partner, offering a comprehensive range of services and tools to support every stage of your journey.
- GitHub Copilot is an AI-powered tool that can significantly boost developer productivity during modernisation by supporting the conversion of legacy code.
- GitHub Advanced Security helps developers respond to growing threats through Secret Protection and Code Security.

Join the AI revolution with Azure

AI transformation through intelligent apps is continuing to accelerate, with **89%** of enterprises planning to increase or maintain their investments in app modernisation, and **61%** prioritising spending on generative AI[1].

Azure is the trusted partner for businesses that need a secure, agile cloud platform for building and managing AI apps – with benefits compared to on-premises infrastructure including:

17%
lower deployment costs[2]

15%
lower ongoing costs[3]

3x
flexibility to iterate and improve[4]

Discover more

To learn more and connect with the developer community, subscribe to Microsoft.Source – our curated monthly newsletter featuring articles, documentation, events and other resources.

Practical learning

- Microsoft App Modernization Guidance for Azure >
- Platform engineering guide >
- Build and modernise AI apps with Azure: reference solutions and content in GitHub >

Big picture

- Microsoft customer success with AI: real-world stories >
- Build and Modernize AI Apps >
- Microsoft Developer >

On-demand webinar

- Secure Development from Code to Cloud with DevSecOps and AI >

Sources:
[1] *Unlock Competitive Advantage With Application Modernization*, a Forrester Consulting paper commissioned by Microsoft, May 2024
[2,3,4] *The Total Economic Impact™ Of Migrating To Microsoft Azure For AI-Readiness*, a Forrester Total Economic Impact™ study commissioned by Microsoft, June 2024