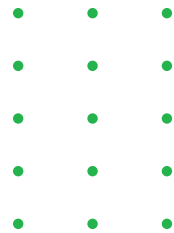
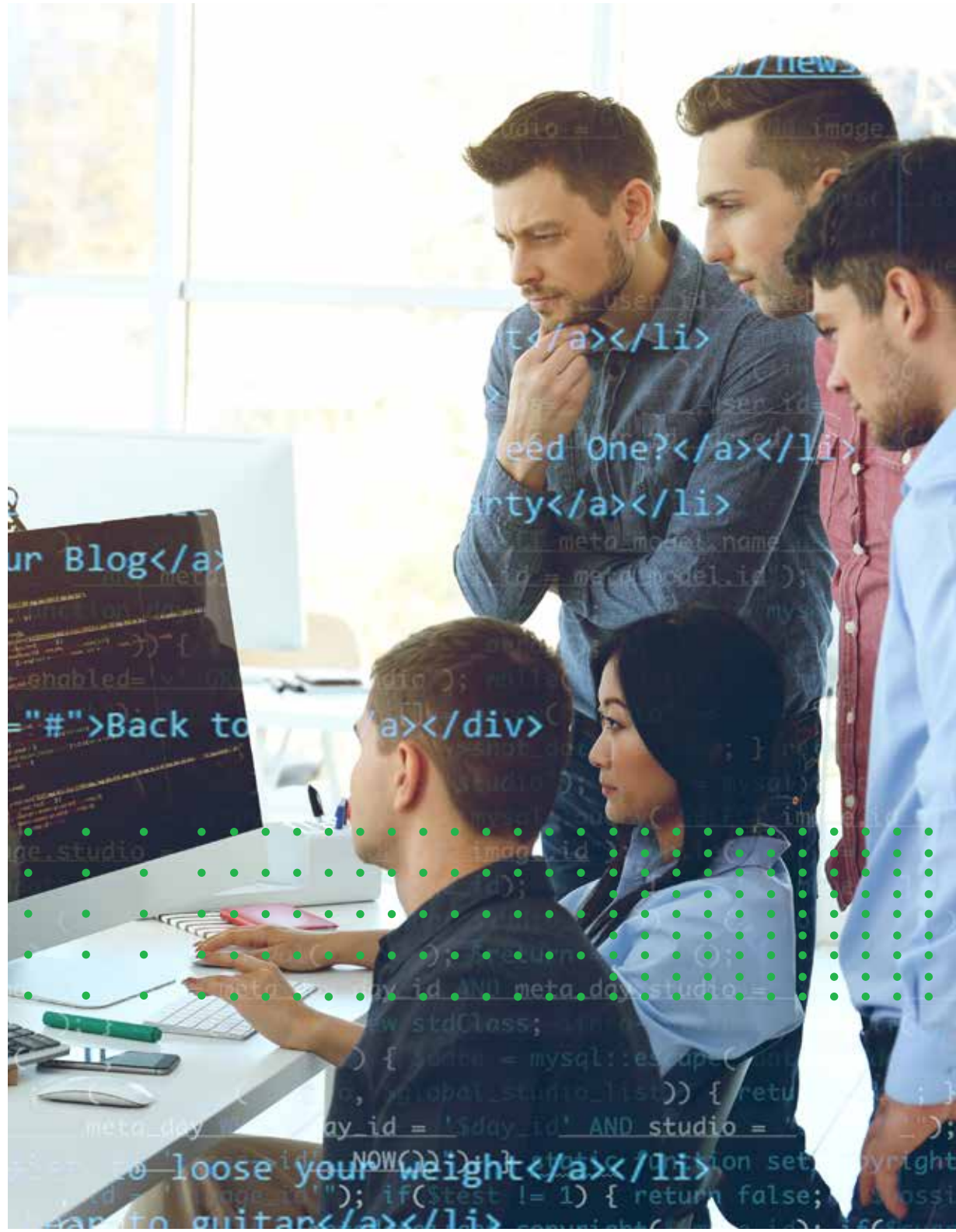


What .NET developers need to know about Azure



APPDYNAMICS



Contents

Microsoft Azure: A History	3
Shifting Perspectives for Microsoft Azure	7
Impact on the Development Life Cycle	10
Introduction to the Latest Azure Innovations.....	14

01

Microsoft Azure: A History



Launched in January 2010, Microsoft Azure is now more than seven years old. Like all seven-year-olds, Azure has been through some memorable teething troubles over the years, as well as some incredible growth and change. In this section, we'll look back at the history of Microsoft Azure, including the ways in which Microsoft's strategy for the platform and for the company as a whole have shifted in recent years. Recently, Microsoft has focused on opening up the Azure environment to widen its appeal to all kinds of developers, including those working in non-Microsoft frameworks and languages.

Microsoft's History From the 80s to the 2000s

First, let's begin with an exploration of Microsoft's strategy during the two decades leading up to the development of Azure. Knowing a little about this period of time is vital to understanding the context in which the very earliest versions of Microsoft Azure were developed.

From the 1980s until the early 2000s, Microsoft's strategy was focused on the Windows ecosystem. During these two decades, Windows grew to become the most popular operating system in existence, allowing both home PC users and businesses to easily interact with their computers.

In the late 2000s, however, Microsoft was in danger of being left behind as the cloud computing revolution started to take off. Amazon Web Services, which remains Microsoft Azure's biggest cloud competitor, launched in 2006, providing online services for websites and client-side applications. Developers were able to use these online services in their applications, which facilitated a more efficient development process and allowed them to build complex apps without having to reinvent the basic frameworks that underpin vital app infrastructure.

Out of the Window, and into the Clouds

Microsoft began working on the development of a platform that could compete with Amazon Web Services in 2008, but it wasn't until January 2010 that Azure finally went live. Since the initial launch, Microsoft

Azure has been through many rounds of changes that have significantly affected the ways in which .NET and other developers interact with the platform.

The shift in strategy has been rocky at times. In 2011, technology reviewers described Microsoft Azure as a confusing platform with a difficult to use web-based interface and incomprehensible documentation. Since then, the teams working on the platform have gradually improved it, resulting in a much more user-friendly platform that offers a well-organized array of services.

One of the biggest changes for the Microsoft Azure platform was its evolution from a Silverlight application to a web portal written in the new HTML5 standard. The new version of the web portal is lightweight and reliable, two properties that make it the platform of choice for many developers.

Microsoft has opened up the Azure environment, adding support for non-Microsoft technologies in order to widen its appeal to all kinds of developers. The platform now supports non-Microsoft databases, both SQL and NoSQL, as well as Linux operating systems. The platform supports developers working in a variety of popular programming languages and frameworks.

In addition to adding support for non-Microsoft technologies, Microsoft has also created a large number of technologies built specifically for the Azure platform. These include storage services, identity management services, and media services.

No longer concentrated entirely on its Windows operating system, Microsoft has renewed momentum and focus steered towards developing cloud-based resources for its developers and businesses, including the Microsoft Azure platform. Cloud computing is now widely recognized as the future of computing as it allows businesses to simplify their IT environments, and developers to draw on libraries and frameworks created by the development community to create highly sophisticated apps.

Today, it's clear that Azure represents an important pivot in the history of the company, allowing Microsoft to become the platform of choice for the world's developers, just as the company's other products have been the go-to option for computer developers in the past.

Competitors to Microsoft Azure

Microsoft isn't the only technology giant that has been working hard over the last decade to develop cloud-based services for developers. Microsoft Azure currently has two major competitors: Amazon Web Services and Pivotal Cloud Foundry.

Amazon Web Services

Amazon was the first major player to enter the cloud computing market with its platform, Amazon Web Services (AWS). In many ways, AWS is still the market-leading cloud-based platform: the Amazon cloud is at least ten times bigger than the cloud capacities of any of its competitors, including Microsoft Azure. However, over the last few years, Microsoft has made a lot of progress in closing the gap between its own platform and AWS. These platforms are the two biggest players in today's cloud computing market by a very wide margin, and professional opinions are divided as to which platform is the best for developers.

The main advantage that Microsoft Azure has over AWS, according to many IT professionals, is that it is more user-friendly, particularly for admins and developers who are used to working in a Windows environment. On the other hand, many experts claim that AWS is the better option for organizations that need to host large-scale applications that support huge numbers of users.

In general, Microsoft Azure makes sense for organizations that are looking for a platform with a simple pricing plan. Microsoft Azure offers straightforward pricing, with tech support billed through a flat monthly fee structure. On the other hand, AWS fees are highly variable: tech support is billed on a sliding scale depending on how heavily you use the platform each month and there are many options for customizing your service plan, which can cause the pricing to vary widely among organizations.

AWS currently dominates the market for Infrastructure as a Service (IaaS), but Microsoft Azure has made progress in eating into this market over the last few years. Microsoft Azure integrates IaaS and Platform as a Service (PaaS) components to provide users with a unified experience that makes it easy for them to start using.

Google Cloud

Recently, technology giant Google has also entered the PaaS space. It's difficult to know how much revenue Google Cloud generates, as the company combines the figures with those for other Google services. However, the popularity of the Google brand could make this cloud platform a serious competitor to Microsoft Azure.

Developers can use Oracle Cloud or Google Cloud as an alternative to Microsoft Azure to build applications. To prevent market share from leaking away to these competitors, Azure needs to continue to reach out to developers.

Pivotal Cloud Foundry

Another cloud-based platform that is worthy of attention is Pivotal Cloud Foundry (PCF). This is the commercial version of the open-source cloud application platform, Cloud Foundry, which is available for free to anyone.

Pivotal Cloud Foundry should not necessarily be thought of as a competitor to Microsoft Azure or Amazon Web Services, though. In 2016, PCF partnered with Microsoft to make their platform available on Microsoft Azure, allowing developers to work on Java and .NET apps with greater ease. PCF made a similar partnership with Amazon, with the result that the platform is now able to run on AWS as well as Azure. Developers can use PCF on Microsoft Azure to build Java microservices-based applications. By working with the functionality offered by Azure, Pivotal provides a continuous delivery platform where developers can quickly develop applications with rich functionalities. Support teams for Pivotal and Microsoft are now integrated, which means you can get support for any aspect of the two platforms during the development and deployment phases of your app.



Looking Forward

Microsoft Azure has a rich and varied history that stretches back over most of the past decade. Even before Microsoft began working on Azure, the company was considering how to make the most of the new cloud computing revolution, and this historical background explains many of the design decisions that went into the early development of the Azure platform.

It's unknown whether Microsoft will be able to close the gap that currently exists between Azure and Amazon Web Services. While there are many reasons to choose Azure over AWS, Amazon's platform is still the go-to choice for a diverse range of well-known brands.

In the following chapters, we'll continue to explore Microsoft Azure, and what it offers you and the industry as a whole, so you can make a more informed decision as you move forward with your cloud strategy.

02

Shifting Perspectives for Microsoft Azure



At the Microsoft Ignite Conference in 2016, Microsoft confirmed its commitment to providing cloud development tools to modern enterprises and developers, using a PaaS approach. This provides enterprises with more options than IaaS, which simply provides infrastructure to enterprises that have apps deployed on their own servers in their data centers.

Microsoft still faces some significant challenges in increasing the adoption of its Azure platform. As Microsoft Azure continues to evolve to meet the needs of modern app development processes and systems, developers will need to stay up to date with the latest changes in order to keep using the platform as efficiently as possible. In this section, you'll learn how to stay up to speed with Microsoft's latest updates to Azure. You'll also find out about possible future challenges that could face developers working on a cloud-based platform such as Microsoft Azure.

Azure's Relationship to .NET

The market still thinks of Microsoft Azure as being primarily a platform for .NET developers. This is largely due to its history, which grew out of Microsoft's desire to provide a platform for people and businesses who were already using its products and frameworks.

Even today, Microsoft Azure's appeal is greatest for those who are familiar with Microsoft tools, such as Visual Studio and SQL Server. For this group of developers, one of the main advantages of Microsoft Azure is its ease of integration of Visual Studio. Developers can use Visual Studio templates to quickly and easily build Azure applications within a familiar environment.

In the future, Microsoft's greatest challenge could be reaching out to developers who prefer to use other frameworks to develop their online and mobile apps. For many of these developers, Amazon Web Services currently seems like a more attractive choice. Microsoft has expanded its platform to support .NET developers looking to leverage other platforms in addition to attracting customers looking to build modern applications with other tools including Node, PHP, and Java, just to name a few.

Microsoft needs to broaden the appeal of Azure to ensure that the platform continues to gain market share. This will involve adding new features to Azure, as well as continuing to work on making the platform as easy as possible for all developers to use, including those who don't come from a strong background of developing in the Windows environment.

Microsoft Azure App Service

Microsoft Azure App Service allows developers and businesses to create and manage four different types of apps, which are Web Apps, API Apps, Mobile Apps, and Logic Apps. Enterprises use all four kinds of apps to interact with their users and allow their operations to run smoothly. They need all these apps to be both reliable and secure. Azure App Service enables developers to build and manage apps within Azure, while taking advantage of the auto scale, auto backup, and recovery features that the platform offers.

Azure App Service is the start of a Microsoft trend toward focusing more on the code and less on the infrastructure, given Azure App Services are still considered PaaS services. The evolution of software development seems to be progressing from IaaS, to PaaS, and now Function as a Server (FaaS).

Azure Functions allows developers to create apps that don't rely on dedicated servers. This means you pay only for the resources your app consumes, rather than having to pay to maintain servers all the time. Azure has embraced this innovative trend and the impact it will have on .NET developers moving forward.

Microservices

Another key part of the Microsoft Azure platform is Microservices, which breaks down applications into a collection of small services, each running in its own environment. This small services approach can make apps more robust and easier to scale, as you can scale each service individually. Microservices helps developers to test each service separately using DevOps tools.

AppDynamics for Azure

AppDynamics gives developers a new set of tools that they can use to monitor their apps. When apps have bugs or issues that negatively affect their performance, developers can use the tools provided as part of AppDynamics to dig down into what's going on in the app and pinpoint the function causing the problem. In addition, Microsoft has also provided support for Xamarin, improving the scope of the platform for app developers who want to target more than one mobile platform.

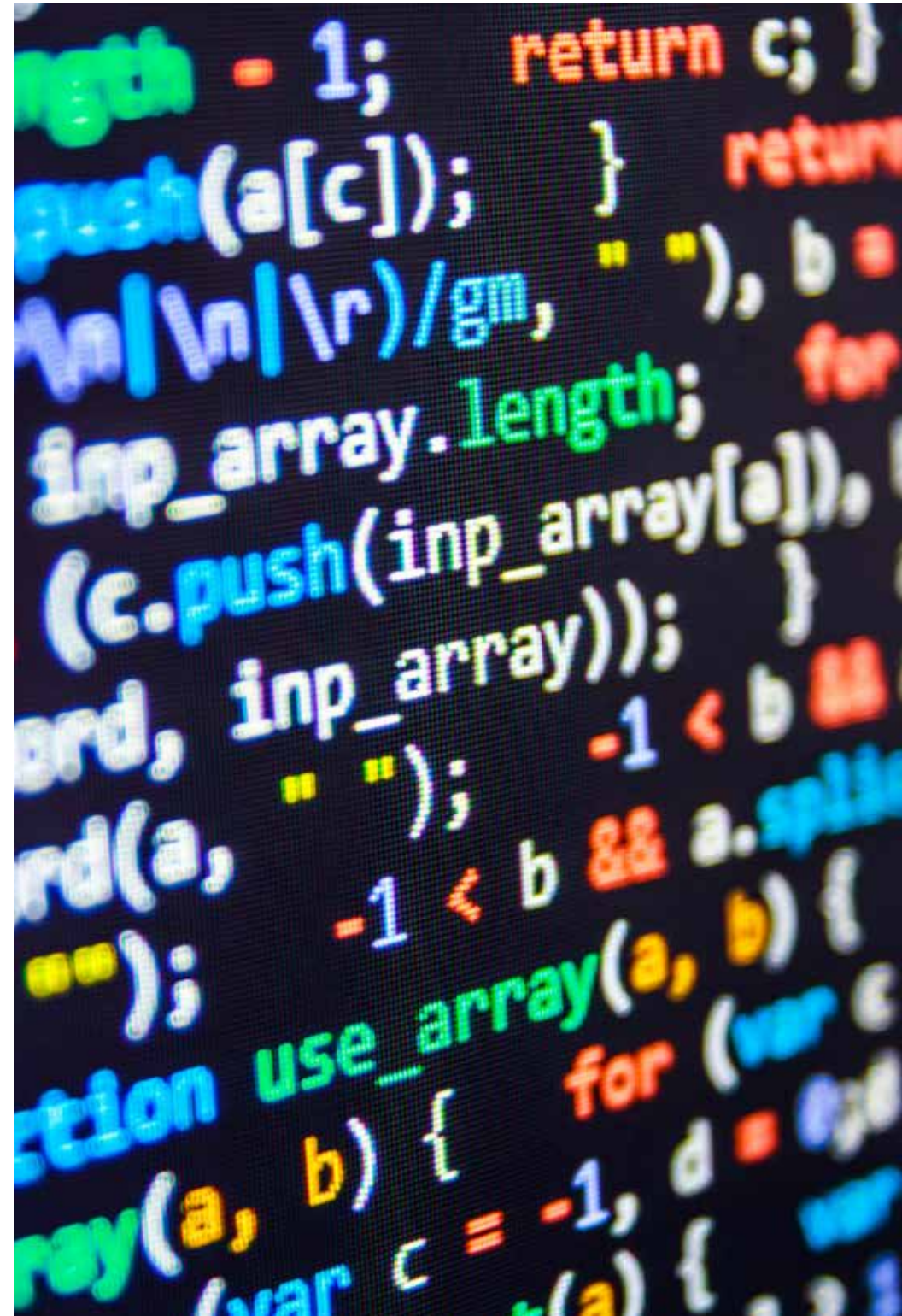
Conclusion

Strongly linked to the .NET framework, Microsoft is making great strides toward widening the appeal of the platform, most obviously through the release of .NET Core, an open source and cross-platform version of the .NET framework.

As a developer, you can make the most of .NET by getting as familiar as you can with all the new features that the platform has to offer. In the future, it's a good idea to stay up to date with the latest news about Azure so you can be sure you always know what updates are in the pipeline. One way to stay current is to read the Microsoft Azure Updates blog and follow Microsoft Azure on the social media platforms you use most often. Attending Microsoft conferences can also help you stay up to speed with the company's latest plans for Azure, allowing you to understand the future direction for app development.

03

Impact on the Development Life Cycle



It's impossible for anyone to deny that Microsoft Azure has had a big effect on the software development life cycle. In this chapter, we'll cover three points for .NET developers that have a significant impact:

1. Leveraging cloud computing with Azure for .NET developers
2. Cross-platform development with .NET Core
3. Xamarin for .NET mobile developers to enable applications to run on any device

Here you'll learn about the effect of the recent .NET Core release, as well as new considerations for app developers working with Azure and new ways they can use the platform to facilitate their app development processes. In particular, you'll read about the impact on mobile developers, who are now able to use Xamarin to build mobile apps for Windows, Apple, and Android mobile devices. All developers are now able to use AppDynamics for performance management of apps, as well as taking advantage of tools that facilitate and accelerate the deployment of mobile and online apps.

Azure's Impact on Application Developers

Let's take a look at some of the most important changes. These include the recent release of a new version of .NET Core, as well as new ways to use REST APIs in Microsoft Azure.

.NET Core

Microsoft Azure offers support for web apps that use the .NET Core platform via Microsoft Visual Studio. To make use of .NET Core in your development work, you need to install the latest Microsoft Azure SDK for Visual Studio, and then install .NET Core and Visual Studio tooling. Once you have signed into your Microsoft Azure account, you can begin working on web apps using the .NET Core platform.

The .NET Core release is a free and open-source web framework that includes a cross-platform runtime implementation of a virtual machine for executing .NET programs and a just-in-time compiler. It provides support for ASP.NET Core web apps, libraries, command-line apps, and Universal Windows Platform apps.

The first version of .NET Core was released in June 2016. However, the release of the latest version, .NET Core 1.1.1 in March 2017, provides developers with more capabilities to develop apps using Visual Studio and Microsoft Azure. To work with .NET Core effectively, .NET developers need to know how to take advantage of computing resources regardless of the platform they are using.

New Ways to Use REST APIs

REST APIs support sets of methods that provide access to resources stored on a server. To use REST requests securely in your application, you need to register your client app with Microsoft Azure Active Directory (Azure AD). You can then define the request URI, HTTP message header, HTTP response header, and HTTP response message body fields in each REST API.

New Considerations

Developers working with .NET Core today can no longer rely on the Windows Event Log to capture events or for logging. Other platforms like Linux and Mac don't have the Windows Event Log. Anyone using Microsoft Azure to support cross-platform development needs to find an alternative event logging tool to monitor their applications on non-Windows platforms.

Leveraging .NET

Application developers can leverage the .NET standard library to create apps that can run in both Windows and Linux environments. The first step for Windows developers is to ensure you have .NET Core tools installed in your Visual Studio environment. It's also possible to install .NET Core on Linux servers, which first requires you to enable the .NET Core channel in your server subscription. You can then get the .NET Core SDK from Microsoft and enable the .NET Core software collection.

Containers

Microsoft Azure offers full support for containers, allowing developers to speed up the life cycle of their software development processes. We'll talk about Azure Container Services in more depth in the next chapter.

Impact on Mobile Developers

Mobile developers need to ensure support for a vast array of platforms, including the various versions of Android, iOS, Windows, and possibly even BlackBerry operating systems. For many developers, Xamarin is the go-to solution for mobile app development, as it allows you to write code that can run on all supported mobile platforms. Developers consider this approach much simpler and more time-efficient than creating stand-alone native mobile apps for each competing operating system.

How Big Is the Xamarin Mobile Developer Market?

In 2016, Xamarin was considered by many mobile application development experts to be a niche solution for a small segment of the market. However, the ease of use that Xamarin offers means it's growing in popularity. Microsoft Azure offers full support for Xamarin, which may encourage more mobile developers to begin using this tool to develop cross-platform mobile apps. Xamarin passed the one million developer milestone in April 2015, according to the Xamarin blog, and its popularity continues to grow. Today, it's estimated that there are roughly 1.5 million Xamarin developers, although information about the exact figure is difficult to find.

Microsoft Azure for Xamarin Developers

For Xamarin developers, the ease of integration between Xamarin and Microsoft Azure is a major draw to use the Microsoft Azure platform. In particular, Microsoft Azure Mobile Services is very useful for this group of people, as it allows them to easily connect a scalable cloud backend to their mobile applications. Structured data can be stored in this cloud and sent out to apps when necessary, using push notifications. Using Microsoft Azure Mobile Services, developers can manage their mobile apps, keep them updated, and send out messages and notifications to users.

Impact of Microsoft Azure on IT Operations

As well as speeding up and facilitating the development of mobile and web-based apps, Microsoft Azure also helps with the deployment and performance management of applications. Let's take a look at some of the key features involved in these crucial stages in the application life cycle.

Impact on Deployment

Microsoft Azure innovations have completely turned the deployment of applications upside down compared to what was happening five years ago. Microsoft Azure now offers continuous deployment for Microsoft Azure App Service apps. Integration of Microsoft Azure App Service with GitHub, Visual Studio Team Services (VSTS), and Bitbucket allows developers to continuously deploy recent updates. This is a great option for app projects that are constantly evolving, as it ensures apps are always kept updated with the latest project developments.

Application Performance Management

One of the most exciting recent innovations related to Microsoft Azure is the release of AppDynamics 4.3. This set of performance management tools allows developers to monitor the performance of their apps and quickly respond to any problems as soon as they arise. This offers more control to developers, as well as ensuring that users have an optimal experience of using the app. The features offered in AppDynamics 4.3 are grouped in the new Developer Toolkit, including: Xamarin support, smart alerts for mobile crashes, and individual user and session crash correlation.

SDLC

SDLC, or software development life cycle, refers to the stages that an application goes through during its development. To create a reliable and stable app, it's vitally important to give full and proper attention to each stage in the cycle. These stages include planning, creating, testing, and deployment. In modern app development, SDLC is also sometimes known as ALM, or application lifecycle management.

Microsoft Azure supports the life cycle management of applications through a series of tools that facilitate app development, testing, and deployment, as well as the performance management that is essential after deployment to ensure that apps are securely delivering the required level of service to users.

Conclusion

In this chapter, we've outlined the impact of Microsoft Azure innovations on the workflows and processes of web and mobile application developers. By integrating various tools and platforms with Microsoft Azure, such as Xamarin and .NET Core, developers can quickly create and deploy cross-platform apps without needing to rewrite code for every platform they want the app to run on. With Microsoft Azure now providing support for Xamarin, cross-platform mobile development is much simpler and more straightforward than in the past, which could partially explain why so many developers are now using Xamarin to help them develop apps for all major mobile operating systems.

04

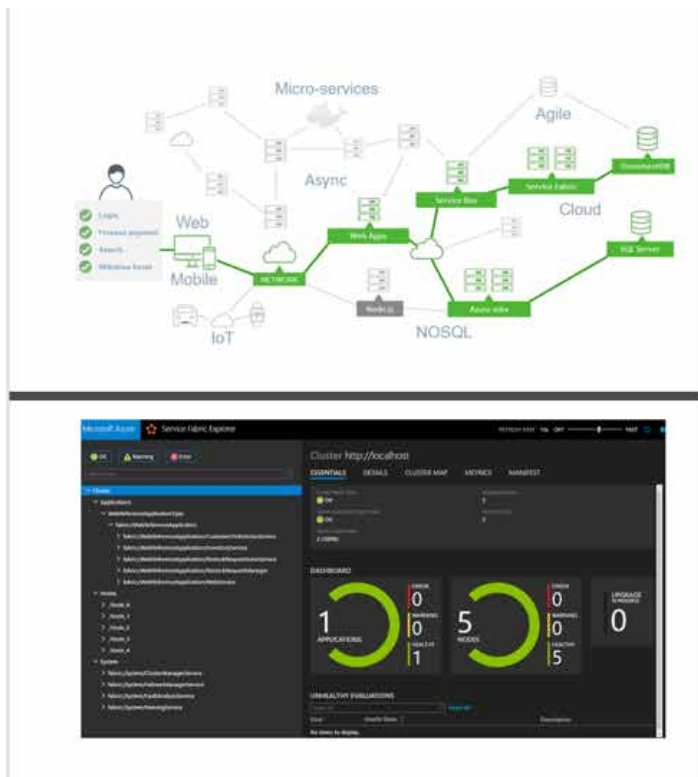
Introduction to the Latest Azure Innovations



In this chapter, we'll take a look at what's new with Azure, exploring the innovations that could change the way developers work across platforms, both on-premises and in the cloud. We'll also dive into the history of Azure and the challenges that exist for .NET developers working in the cloud.

Top Azure Innovations

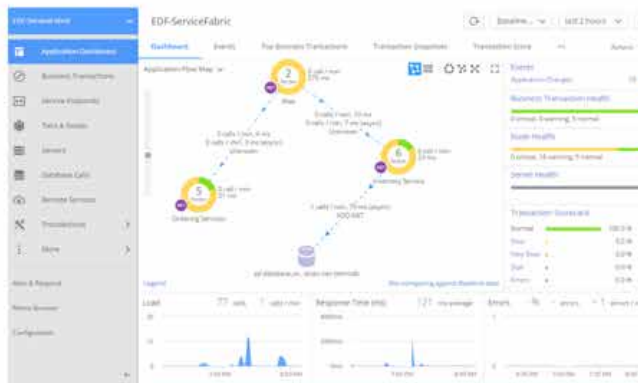
Microsoft is constantly updating the Azure platform to make it more useful for .NET developers as well as developers across the ecosystem, including Java, Node.js, and other modern runtimes. Some of the most useful innovations include Service Fabric, Azure Container Services, and the ability to develop mobile apps in a cross-platform framework. Let's take a look at each innovation to understand its impact on the future of .NET development.



Micro-services With Service Fabric

Azure Service Fabric is a platform that allows you to easily package, deploy, and manage the services associated with your applications. It's designed to reduce the significant challenges that have faced developers managing large-scale cloud applications in the past. In particular, Azure Service Fabric is designed with scalability in mind, allowing developers to scale up their apps without sacrificing reliability or manageability.

Using Service Fabric, you can build and manage applications that are composed of a number of microservices that run on a cluster of machines. This approach is extremely reliable. It also allows you to scale various parts of your application in turn, according to the demands of your users. Service Fabric encourages developers to apply the microservices design pattern within their app development for more agility and componentization, rolling out new features and updates frequently to meet business and customer demands.



Azure Container Services

Containers are a key part of modern software development, maintenance, and delivery. You can use containers to simplify the app development process and avoid rewriting configuration scripts between environments. For many development teams, adopting containers is an important aspect of embracing an agile approach to software development. Azure Container Services allows developers to create a container hosting service that is optimized for their existing Azure development environment. It's possible to migrate container workloads into and out of Azure without making changes to the code.

Microsoft Moves Cross-Platform

Microsoft Azure not only supports developers who work in a Windows environment, but also those who need to work with the Linux and macOS platforms. As mentioned in the previous chapter, Xamarin helps developers create apps that work across all platforms, ensuring a consistent user experience and reducing the burden of maintaining a completely different version of your app for each platform.

Cross-Platform Mobile Development

Developing mobile apps for a range of platforms, including iOS, Android, and Windows, allows developers to reach out to a wider audience of mobile users. With its latest round of developer platform updates, Microsoft shows its support for mobile development with the goal of allowing developers to target any device regardless of platform with the acquisition of Xamarin and development of the Azure Mobile App Services. This is vitally important for developers who need to reduce the hassle associated with maintaining apps for different platforms.

Cloud Computing for Deployment and Monitoring

Microsoft has taken steps to make deployment and monitoring as quick and hassle-free as possible for .NET developers. Azure is strongly positioned toward cloud developers, allowing them to collaborate on development projects on a centralized, cloud-based platform.

Azure App Service and Serverless Computing

Azure App Service provides platform-based services that continue to push the infrastructure into the background, allowing developers to focus on the code needed to build solutions instead of the infrastructure used to run the code. Azure App Service includes Mobile Apps for backend mobile services like notifications, Web Jobs for running batch jobs, API Services for developing and hosting Web APIs, and Web Apps for hosting front end web applications.

Microsoft has introduced its own offering into serverless computing with Azure Functions. Azure Functions allows the developer to focus on writing and deploying the code required to respond to events produced by the actions of their customers or other integration points.

What's New in AppDynamics 4.3?

AppDynamics has several new features. For .NET developers, the two most exciting are the new Dev Toolkit and Xamarin Support.

Dev Toolkit

The new developer toolkit contains tools that help .NET developers create applications that work across a variety of platforms. These help developers to securely use containers in their development with greater ease than ever before. Using the application performance monitoring tools available for apps written using a variety of languages and frameworks, including Java, Node.JS, PHP, and .NET, developers can monitor the performance of their apps to ensure a positive experience for all users at all times.

Xamarin Support

AppDynamics provides full support for Xamarin. The Xamarin test cloud has the potential to be particularly useful for developers, as it allows you to test your apps from the point of view of a user — each test performs taps, pinches, swipes, and other simulated finger movements to check that the app performs as expected, no matter what the user does.

AppDynamics and Azure Language Stacks

The innovations associated with the release of AppDynamics 4.3 for Microsoft Azure could hold great potential for developers working with a number of programming languages and frameworks. Let's take a look at the features that can be useful for the most important programming languages and frameworks in use today.

Support for Multiple Programming Languages

One of the main advantages of Microsoft Azure is that it's cross-platform and language agnostic. That means programmers can work in whatever language they're most comfortable working in, while still taking advantage of all the features offered by Microsoft Azure and AppDynamics.

As a .NET developer, multi-language support can be vitally important to your working processes. It allows you to engage with other teams who may be using a different programming language to deploy their services. You don't have to worry about getting stuck in a single-language rut — Azure's wide support for multiple languages allows you to work with a wide range of people.

AppDynamics provides the ability to monitor and measure app performance across the entire stack. This allows developers to monitor the performance of applications written in a range of languages, including those that operate in complex and distributed environments. The application performance management tools offered by AppDynamics focus on tracking key business transactions, which can be automatically discovered using the AppDynamics tools.

Microsoft Azure supports the following languages: Java, PHP, Node.js, Python, C++, and many others.

.NET

For .NET developers, one of the most useful innovations offered by AppDynamics is the enhanced ability to offer .NET application performance. As with Java, Node.js, and PHP apps, this monitoring is focused on Business Transactions, which can be tracked through complex and distributed environments to give a comprehensive picture of an app's real-time performance.

When you use .NET monitoring in AppDynamics, you can deploy agents in three roles (web, worker, and CustomVM) to give you as much visibility as possible into the performance of your .NET Azure services. Tools in AppDynamics' .NET application performance management also allow you to visualize all the dependencies of the application components that interact with your CLR. This high-level view of your application allows you to identify the parts of the app that are likely to be causing problems in overall performance.

Conclusion

Microsoft Azure has played a significant role in shaping the landscape for developers. By allowing them to develop in a cloud-based platform, Azure supports a modern, efficient development process.

With the introduction of AppDynamics 4.3 and its range of useful features for .NET developers, Microsoft has shown its commitment to supporting a cross-platform approach to application development. No matter what language stack you use – Java, PHP, Node.js, .NET, or some combination of several stacks – you can use AppDynamics to manage and track the performance of your apps within Microsoft Azure.

About Edward Ferron

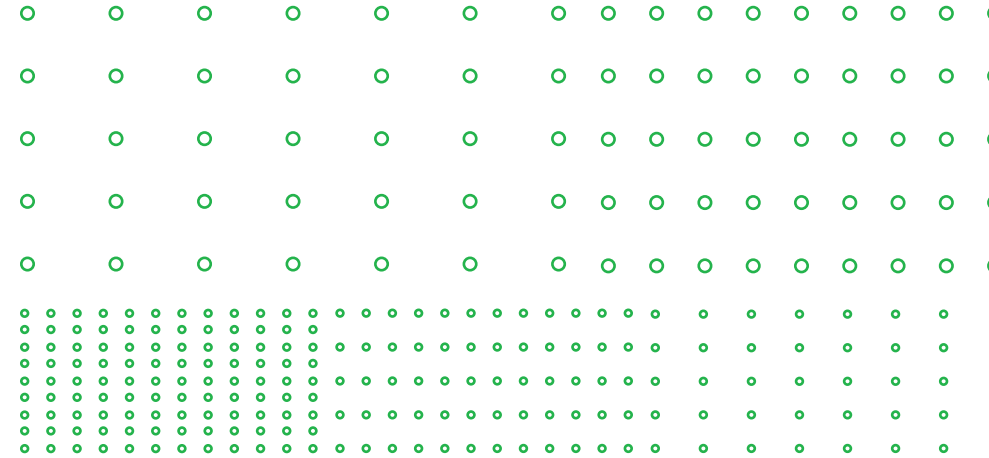
Ed is a Solutions Engineer helping customers meet their web-scale demands with production performance monitoring from AppDynamics. Ed joins AppDynamics from Microsoft where he served as a Principal Architect helping customers implement enterprise applications. Soon after, Ed joined the Microsoft Technology Center as the Technical Director in the Midwest, where he conducted Architecture Design Sessions and Executive Briefings for customers across North America. A Software Architect by trade, Ed enjoys publicly speaking to share his expertise in Azure, web technologies, and helping shift large-scale enterprise software solutions to take advantage of modern cloud computing capabilities.



Sources:

<http://garyclarke.us/technology/a-brief-history-of-microsoft-azure/>
<https://www.slideshare.net/MattDeacon/a-brief-history-of-azure>
https://en.wikipedia.org/wiki/Microsoft_Azure#Timeline
<https://pivotal.io/microsoft>
<https://www.cloudcomputing-news.net/news/2016/sep/06/aws-us-azure-it-pros-weigh-pros-and-cons/>
<http://www.techrepublic.com/article/aws-now-10x-the-size-of-its-competitors-is-the-cloud-arms-race-over/>
<http://www.computerweekly.com/feature/Does-Microsoft-Azure-make-sense-for-developers>
<http://searchcloudcomputing.techtarget.com/feature/Azure-adoption-gets-a-boost-but-challenges-remain>
<https://azure.microsoft.com/en-us/blog/topics/updates/>
<http://blogs.perficient.com/microsoft/2016/09/the-future-of-azure-paas-cloud-dev-for-the-modern-enterprise/>
<http://www.informationweek.com/cloud/infrastructure-as-a-service/boeing-win-positions-azure-for-future-enterprise-growth/a/d-id/1326322>
<https://docs.microsoft.com/en-us/aspnet/core/tutorials/publish-to-azure-webapp-using-vs>
<https://docs.microsoft.com/en-us/rest/api/>
<https://www.microsoft.com/net/core#linuxredhat>
<http://www.mindgrub.com/blog/saad-raouf/review-cross-platform-development-xamarin-1>
<https://weblogs.asp.net/scottgu/announcing-windows-azure-mobile-services>
<https://docs.microsoft.com/en-us/azure/app-service-web/app-service-continuous-deployment>
<http://searchcloudcomputing.techtarget.com/definition/Windows-Azure>
<https://azure.microsoft.com/en-us/features/xamarin/>
<https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview>
<https://blog.appdynamics.com/news/appdynamics-interviews-container-expert-liz-rice/>
<https://www.appdynamics.com/java/>
<https://www.appdynamics.com/php/>
<https://www.appdynamics.com/nodejs/>
<https://www.appdynamics.com/net/>

START A FREE TRIAL AT
APPDYNAMICS.COM



About AppDynamics

AppDynamics is the Application Intelligence company. With AppDynamics, enterprises have real-time insights into application performance, user performance and business performance so they can move faster in an increasingly sophisticated, software-driven world. AppDynamics' integrated suite of applications is built on its innovative, enterprise-grade App iQ Platform that enables its customers to make faster decisions that enhance customer engagement and improve operational and business performance. AppDynamics is uniquely positioned to enable enterprises to accelerate their digital transformations by actively monitoring, analyzing, and optimizing complex application environments at scale which has led to proven success and trust with the Global 2000. For more information, visit appdynamics.com.

APPDYNAMICS

appdynamics.com