

05242022 Build Scott Guthrie

**Build 2022**  
**Scott Guthrie**

**SCOTT GUTHRIE:** Thanks, Kathleen. Hello, everyone, and welcome to Microsoft Build 2022.

Developers are essential to the world we live in today, and the work you do is critical to powering organizations in every industry. And with the Microsoft Cloud, you have a trusted and comprehensive platform that enables you to build great apps and solutions.

The core of the Microsoft Cloud is Azure, our cloud computing platform and the underlying infrastructure and platform that enables you to run code. We also provide advanced SaaS-based solutions that provide productivity services, business applications, low-code tooling, and advanced security and identity management capabilities that you can leverage as well. And with Visual Studio, .NET and GitHub, we provide the world's best and most popular development tools. We're going to talk more in this session about how you can use the Microsoft Cloud and all of these capabilities to build great cloud-native apps.

As you build new apps, you're going to want them to be cloud native. Cloud-native apps are capable of delivering new levels of scale and performance and can provide even greater reliability for you and your customers. Using cloud-native design patterns helps achieve the agility, efficiency and the speed of innovation that you need to deliver for your businesses.

Tens of thousands of customers are already building cloud-native apps on Azure today. This includes organizations like Walmart, Starbucks, ASOS and the National Health Service in the U.K. Using cloud-native design patterns and Azure has helped each of these organizations achieve the agility, reliability, scalability and security demanded by their businesses and end users.

The experience bar is going up in terms of what end users expect from apps, and you never know when tens of millions of customers might show up suddenly. Things like product launches, peak shopping seasons and sporting events are just a few examples of the types of highly dynamic usage demands that modern apps must address.

And one of the great things about cloud-native apps is the ability to quickly scale up to meet this demand and handle huge amounts of load, everywhere around the world. Cloud-native applications take advantage of containers, serverless technology, microservice-based architectures and API-based solutions to enable you to build and iterate solutions faster. Cloud-native apps are delivered using a highly automated dev ops-based approach, enabling you to continually deploy new updates and capabilities to

your customers. And with Azure, GitHub and the Microsoft Cloud, we're working to enable you to easily leverage all of these capabilities and approaches.

We know that many of you are in various stages of your cloud-native journey. You often have existing investments, whether those are in the existing tools you use, the code you've already written, or even across other clouds that you take advantage of. And with the Microsoft Cloud in Azure, we meet you where you are and make it easy for you to start your cloud native journey from anywhere.

Let's look at one customer that's been through this journey of adopting cloud native.

Now, H&M is a global retailer that had over 750 applications to modernize, many of which were built as Web apps connecting directly to databases. H&M looked to Azure to build and deploy their cloud-native solutions and to start containerize their applications. They then focused on designing and building an event-driven application model and exposing APIs using a microservice-based architecture. And this ultimately gave H&M a more flexible and scalable order fulfillment solution that better met demands of inventory sourcing and order management. And it's enabling them to bring new capabilities to market much faster than ever before.

Now, as you go down your own cloud-native journey, you can start to modernize your code using Azure Container Apps. Azure Container Apps as a serverless, container-based hosting service that allows you to stay focused on your apps, instead of having to worry about managing the underlying cloud infrastructure. It supports event-driven scale. In fact, you can even scale down to zero instances. If you're not using it, you're not paying anything for your apps running on it. And if you then suddenly get a lot of incoming requests, you can quickly and dynamically scale up to many thousands of instances to handle that load.

It encapsulates microservices best practices, such as secure service-to-service communication and reliable messaging through pub sub. And to show why companies like H&M are choosing Azure Container Apps to accelerate their cloud-native strategy, I'd like to show you a demo.

In this demo, we'll use a company called Contoso Air, who is modernizing one of their workloads that manages customer bookings, and they're enhancing it with cloud-native benefits around scale reliability, security and agility through microservices.

Let's start with the basics. I've got a terminal here connected to an empty Azure subscription. You can see right now, I don't have any containers or anything running. I'm literally starting from scratch in Azure.

Now what I want to do is get this code, which is currently checked into a repository on GitHub, running in the cloud. This is a JavaScript-based server application, and what I want to do is to get it up and running as a microservice in Azure. And to get this deployed and running at cloud scale, I just need to type single command, easy container, up. With

this command, I'm going to point it to the GitHub-based repository and give it a name, and that's it.

Now I'm securely connecting my Azure account to my GitHub repository. Azure is now my serverless container environment that's going to run the app, and this command is creating for me an Azure Container Registry to securely manage my container images. And it's automatically creating and configuring a GitHub CICD action that will build my code, run my tests, and then deploy the built app into Azure Container Apps to run.

And it set all of this up with the CICD dev ops workflow, which means that at any time I want to make a change or improvement, I can simply open a pull request or commit the code into the branch, and GitHub will automatically build, test and push the update up to Azure.

There you go. The GitHub action has completed its build, test and deployment, and now, I have that code running as a serverless container in Azure. One command and my repo is now running as a cloud-native app.

Let me show you a little of what I now get from Azure Container Apps from a runtime and management perspective. Now I can continue to manage and configure these apps directly from the command line, but let's jump over to the Azure management portal and instead manage it that way.

Now one thing I want to highlight is the serverless scale you get with Container Apps. You can see here, I can configure my container to scale all the way down to zero instances when there's no traffic or events. What's great about this is when my container scales to zero, so does my bill. I'm not paying for anything when it's at zero. I can also configure real-time scale, though, based on rules including incoming HTTP requests, as well as have my containers scale in response to things like queue messages, event streams or even database events.

And I've now got everything I need to manage and monitor my apps. I can configure custom domains, certificates and secrets. And you can see I can even configure and manage that secure connection between my GitHub actions and this app right from within the Azure management portal or command line interface. And I can now repeat this process with more of my code bases, deploy multiple containers to the serverless environment to create my modern and cloud-native solution.

And behind the scenes, this is all powered by our Azure Kubernetes Service and open source technology, like DAPR, Kata and Envoy, which gives me confidence that if I ever need more control or flexibility, I can move this solution to our full Azure Kubernetes Service for the full power of Kubernetes.

With Azure Container Apps, we've made cloud-native and microservices-based development streamlined and productive for any developer with any experience or skillset. We're seeing great success with our public preview, and today, I'm excited to

announce Azure Container Apps is now generally available and ready for everyone to use. We're really looking forward to seeing the great apps that you build with it.

Now, Azure Kubernetes Service enables you to take full advantage of the Kubernetes ecosystem and scale your cloud-native development even further. A great example of a customer taking advantage of this today is Adobe.

The Adobe Document Cloud includes Adobe Acrobat, document APIs and Acrobat Sign, a leading cloud-based, e-signature service used across regulated businesses, governments and institutions around the world.

Adobe evolved to cloud-native practices a few years ago and adopted a microservices architecture that utilizes Azure Cosmos DB, Azure Cache for Redis and Azure Kubernetes Service. Adobe chose Azure Kubernetes Service because of its scalable, flexible and multicloud capabilities. This enabled Adobe to speed up development from onboarding to production, all while providing automated guardrails with dev sec ops practices.

Azure Kubernetes Service delivers elastic provisioning of capacity without the need to manage the underlying compute infrastructure, and it's the fastest way to spin up managed Kubernetes clusters and configure a seamless dev sec ops workflow with CICD integration.

And we have several great updates that we're making at AKA this week at Build. To help developers be even more productive with Kubernetes, we've integrated the open source draft project to streamline getting started building apps that run on Kubernetes. This draft integration allows you to take a non-containerized application and generate the Docker files, Kubernetes manifests, helm charts, customized configuration and other artifacts associated with a containerized solution. We're launching the draft extension and CLI that makes it super simple to use and leverage the full Microsoft ecosystem, from Visual Studio Code to GitHub to Azure.

The several other AKS capabilities are also now in public preview. We're announcing a new integrated AKS Web application routing add-on that enables quick, secure access that sets up an ingress controller, configures DNS as well as SSL, TLS authentication.

KEDA is a popular open source project that handles auto scaling. With the KEDA AKS extension enabled, AKS clusters can now be scaled down to zero pods, decreasing consumption when not needed. And then, when new requests arrive, KEDA activates the deployment to answer the incoming traffic.

Also now in public preview today is our new Azure Managed Grafana service. We know observability is critical, and this preview allows you to deploy familiar Grafana dashboards, complete with built in high availability and security right from within Azure. And I'm excited to bring Brendan Burns into the discussion to go into a demo of AKS and some of our announcements.

**BRENDAN BURNS:** Thanks, Scott. I'm going to start with a Contoso air demo that's up there on GitHub.

Now the first thing is to set up a code space. Codespaces on GitHub is a really fantastic cloud-based development environment that's entirely in your browser. I'm here in a familiar environment, basically VS Code, but everything that I need is up in the cloud and running inside my browser.

Now I'm going to run AKS Draft to create my configuration. You can see here, it's taking care of everything for me, even installing binaries that I might need, detecting the language that I'm programming in, and asking me in some instances for information like the port that I want to run on. I'm going to choose Helm as my deployment type. And again, I'm going to expose the application on Port 8080. I'll give it the name Contoso Air, and now I'm ready to go. Everything I need to containerize this application into AKS has been generated for me.

Now I'm going to choose a GitHub action-based workflow to deploy that application out to my AKS cluster. Again, it's going to prompt me for some information that it needs. I'll enter all of that in, including the AKS cluster that I wanted to play my application into. And once I have that information into Draft and the workflow, you can see that it's generated a GitHub-based workflow that's going to do the deployment.

Now let's see that workflow in action. We're going back over to Codespaces, and I'm going to commit all of that code that's been generated for me. You can see there's a bunch of busywork that's been taken care of for me, and it'll follow Azure's best practices for application deployment also. I'm going to commit all these changes and push them out to my GitHub repo.

When you look at GitHub, you can see that the commit is there. But more importantly, what you're going to see here is that there's an action that's been started. All that code that was generated for you by the Draft tool is actually busily taking care of deploying this application.

If you look at my cluster in the Azure portal, you'll see that there's already Kubernetes resources created by this release, including the external address of my Contoso Air application, from code to Kubernetes in just a few commands.

Now that I'm on Contoso Air, I'm going to buy a ticket to Hawaii. It's been a little chilly here in the Seattle area.

Obviously, deploying an application is great, but for real-world applications, it's critically important that you monitor that application. That's why I'm so excited about the new Managed Grafana service that we have available for you within Azure. You can see that all it takes to install the Grafana service into the same resource group as AKS is just a

few commands and a few seconds, and you'll have that cluster available to you to monitor your application.

If I go back over to the Azure portal and I go into the resource group that my AKS cluster is located in, and now there's a managed Grafana instance also. And if I go into that Grafana instance, I can see all the metrics and data provided by container insights, but this time visualized in the fantastic and familiar interface of Grafana. You can see information like requests for seconds, the resources that you're using for your pods, and all of the different things you need to run your application successfully.

In a sense, Grafana is an open source product. It's available to you in Azure or wherever you want to run it.

It's exciting to see how easy Azure can make running an application on top of industry standard, open source software. But it's doing it in a managed way so that you can focus your energy on the applications that run on top of that software.

Back to you, Scott.

**SCOTT GUTHRIE:** Thanks, Brendan. With the Microsoft Cloud and Azure, you can build upon and leverage a complete cloud-native platform, from containers to serverless to cloud-native databases and AI capabilities. Azure Cosmos DB is the only petabyte-scale database service in the world that offers SLAs for single-digit millisecond latency and five-nines availability with turnkey georeplication capabilities. This enables you to build incredibly large and sophisticated cloud-native solutions.

We're improving our Cosmos DB serverless capabilities and now enable you to dynamically scale up and down your database scale, while only being charged for the storage and throughput that you use. The serverless, consumption-based database operations mode makes it much more cost effective to run both dev test workloads and new production applications, and is a great way to get started using Azure Cosmos DB.

With Azure, we provide a rich set of fully managed cloud services that you can use. These fully managed cloud services provide capabilities like high availability, automatic patching, backups, security management and more. And with Azure Arc, we now make it possible for you use these managed cloud services literally anywhere, whether it's in Azure, multicloud or hybrid environments and at the edge. And this now allows you to modernize even existing on-premises applications to be cloud-native solutions, using familiar tools and Azure services.

I'm excited to announce that our business critical tier of our Azure Arc-enabled SQL managed instance is now generally available as of this week. This enables you to meet the most demanding business continuity requirements and gives organizations access to top-rated security and automated update capabilities.

Azure Arc-enabled App Services are also now in preview. These enable developers to take advantage of more of our cloud-native application development capabilities everywhere, including in hybrid and multicloud environments. You can now run our Azure App Service, functions, logic apps, API management, event grid services on any Azure Arc-enabled Kubernetes cluster. Then the Kubernetes cluster can be on premises or hosted even in a third-party cloud.

Now, we've covered a lot of territory so far, watched some amazing demos and announced some new services. Developing with the Microsoft Cloud puts the latest cloud technologies in your hands and empowers you with both control and productivity.

In the next section, you'll hear from Julia Liuson and Ryan Morgan from VMware about modernizing enterprise job applications. As always, I can't wait to see the amazing solutions you build.

Over to you, Julia.

**JULIA LIUSON:** Thanks, Scott. I'm delighted to be joined by Ryan Morgan from VMware to talk about how we are helping Java developers run their sprint applications in the cloud.

Ryan, can you talk about how we started working together?

**RYAN MORGAN:** Thanks, Julia. Microsoft and Pivotal, now VMware, have been partners for a long time on Spring. In 2019, we announced Azure Spring Cloud, a fully managed service for Spring Web applications that solves the challenges of running Spring at scale.

Since then, we've expanded the capabilities of the service, including the enterprise tier that we announced in preview at Spring 1, 2021. Azure Spring Cloud is a fully featured platform for all types of Spring applications, and to better reflect this, we are now calling the service Azure Spring Apps.

**JULIA LIUSON:** A great example of how a customer is using Azure Spring apps to run their Java apps at massive scale is FedEx. They're leveraging the Microsoft and VMware collaboration on Azure Spring Apps to deliver an impactful solution for their end customers.

FedEx Global Delivery Prediction Platform, or GDPP, is a Java based solution running on Azure that helps predict estimated delivery time for millions of packages. GDPP dynamically predicts estimated time of arrival for shipments, which has great benefit for customers with time-sensitive deliveries in industries like manufacturing, healthcare and construction.

I'm really proud of what we have built with Azure Spring Apps and the ability to support customers like FedEx in driving business outcomes to continue to support our customers

running mission-critical Spring apps in the cloud. Today, we're announcing the Azure Spring Apps Enterprise will be generally available in early June. This tier builds on the value of basic and standard tiers and has fully managed VMware Tanzu components running on Azure, and advanced VMware Spring runtime support.

Ryan, what are some of the really cool things developers can do with Enterprise?

**RYAN MORGAN:** One of the key goals of Enterprise was to make it easier for developers to focus on their apps instead of dealing with operational complexity. With Enterprise, they can select and configure VMware Tanzu components of their choice at instance creation.

Tanzu Build Service and Tanzu Buildpacks automate source-to-container workflows across all your development frameworks. This includes the ability to deploy polyglot apps alongside your Spring apps. Here, you can see Buildpacks being applied through GitHub Actions.

Next, API portal for VMware Tanzu, along with Spring Cloud Gateway for VMware Tanzu, enables you to publish, discover and route application traffic. Any APIs in Spring Cloud Gateway are automatically surfaced in API portal for easy discovery and testing. Additionally, Enterprise tier includes VMware Spring runtime support, which gives you access to Spring experts and a minimum of 12 months support after OSS end of life.

**JULIA LIUSON:** Thanks, Ryan. It's really exciting to see all of this and we'll continue to partner to make Azure Spring Apps even better.

END