

09252017 Ignite Technology Keynote Scott Guthrie

**Microsoft Ignite
Technology Keynote
Scott Guthrie
September 25, 2017**

ANNOUNCER: Please welcome Executive Vice President Microsoft Scott Guthrie.

(Applause.)

SCOTT GUTHRIE: Good morning, everyone, and welcome to Ignite.

This morning Satya demonstrated some of the incredible innovations coming in the future and the amazing impact that technology is going to be able to have on people's lives. I'm now going to go deeper into Microsoft Azure and how you can take advantage of it to build amazing solutions.

One of the defining aspects of cloud computing is the ability to innovate and release new technology faster and at greater scale than ever before. There's a set of technology, things like IoT, AI, micro services, serverless computing and more, this is all happening right now thanks in large part to cloud computing. And it's an incredibly exciting time to be in IT and the opportunities for new approaches and new technologies have never been greater.

At the same time I also recognize that while all this stuff is really cool, it can also be a little overwhelming. I hear that in a lot of my conversations with customers around the world. The expectation to know all of these new technologies and be up to date with them all the time can sometimes leave you feeling like you're falling behind. And the expectations that companies have on you to quickly deliver breakthrough experiences with all this new technology is super high. In a lot of cases, your companies are betting on you really to deliver the new digital experiences that are going to transform and power those businesses going forward.

At the same time this constant worry we all have about someone hacking the apps and systems that we're building. And you know you have to be super careful about security while simultaneously trying to be an expert on all this new tech and deliver breakthrough solutions on schedule. And needless to say all of this is not easy.

And it's really with this understanding that shapes how we build and deliver Azure. My team comes to work each day working to make Azure a powerful enterprise-grade cloud service. But the more important work we do is building Azure to help guide your success. Having great technology and lots of features is necessary but not sufficient. It's really about how successful you can be using this technology and the cloud.

And to deliver on this we focus Azure innovation really on your needs. By making cutting-edge technology approachable to all IT professionals and doing the heavy lifting to ensure that Azure uniquely meets enterprise scenarios. This means having an end-to-end experience across all of our cloud services, our management tools, our development tools, that provide an incredibly productive cloud experience, one that's hybrid and that enables you to build solutions that run consistently, both inside our public cloud data centers as well as within your own data centers. It means having a cloud that enables you to use AI and data to infuse richer intelligence into your solutions, and it means having a cloud that you can trust. Trust is a core value of Azure, and we lead the industry with our work on security, compliance, privacy and responsibility.

This focus that we have on delivering innovation, trusted results, is really leading to tremendous adoption of Azure right now. Startups, governments, and over 90 percent of Fortune 500 companies in the world are now running their businesses using the Microsoft Cloud. And these are just a handful of some of the customers running on Azure today.

What I thought I would do is just show a quick video of some of them talking about how they're using Azure to drive their success. Let's watch a video.

(Video segment.)

Now as you move increasingly large and complex applications to the cloud, you need comprehensive tooling to build, deploy and manage them efficiently. And one of the things that really makes Azure unique is the end-to-end management experience that we deliver, whether it's with our management portal, our new PowerShell and batch command line experiences, and with the built-in support for making things like monitoring, log analytics, patching, backup and site recovery super easy.

And to give you an idea of kind of what this overall looks like, we'll just show off some of the great new capabilities that we're releasing today. I'd like to invite Corey Sanders onstage to demo Azure in action. Here's Corey.

(Applause.)

COREY SANDERS: Excellent. Excellent. Good morning, everybody. I am going to show you how easy it is as an IT professional to be able to use Azure. I'm going to make you more efficient, more productive and hopefully have a little more fun in the meantime. So I'm really excited. Are you guys excited? (Audience response.) All right.

So I'm going to start with infrastructure. We have lots of infrastructure in Azure. We have lots of different options. We have GPU-powered infrastructure. We have InfiniBand-powered HPC machines. We have large CPUs. We have lots of memory. What I'm going to show you today is our really, really large machines, our M Series.

And so here I've got 128 vCPUs. I'm going to zoom in here and show you, here I'm running at 3.8 terabytes of memory. So this is a honking machine. And this is going to

be good for running very large SQL Servers or SAP Hana to be able to run on this. But you can also do some other interesting things on this. You can run nested virtualization right on top of this. So here you can see I'm running two virtual machines inside this virtual machine. So I'm running a Linux machine and a Windows machine inside. And so you may want to do this for Hyper-V replica or taking Hyper-V snapshots. You can also do Hyper-V-based containers to get higher security-based containers, all running inside this machine.

And what's great about this is, I'm running both Windows and Linux in here. If I actually click open my Windows machine you'll see another running Hyper-V. So now I've got another Hyper-V inside a virtual machine. So this is a VM inside a VM inside an Azure VM. So it's truly an inception moment here for all of us. And, of course, running this is quite a dream.

Let's go back. So I've shown you how to create VMs a lot in the past. But what I'm going to show you today, instead of creating a VM through the portal, I'm going to show you a different way. I'm going to show it using PowerShell right here in the portal experience. You can see here available today in preview PowerShell built into the Azure portal.

So this goes anywhere. This is tied to the browser. (Applause.) Tied to your browser, can run on any OS, it can run in any browser, and it can also run on your iPhone. So you can start doing PowerShell right on your iPhone.

And let me show you, if you want to create a virtual machine here, if you're familiar with our PowerShell it used to take many, many commands, many, many commands to get this going. Now it takes just one parameter, and with that I'll pass in my username and password and it creates a VM. So you don't have to worry about all the other configuration unless you want to, just that fast.

Now if I want to go in and actually run a query, if I want to actually go do some searches, I can also do that right here in this experience, and so if I'm actually going to go in and run, let's say I want to get all my virtual machines running here, and then let's say I want to actually stop them. I'm done with my demo. So let's go ahead and stop them. But, wait, I'm a little bit nervous. I don't know all the VMs that are in my account.

So I can use classic PowerShell "what ifs" to validate what this command is actually going to do. And you can see it's going to run through and tell me what will happen if this command runs. So available right here as part of the PowerShell commands and you can see as I go through this that there is a VM down here that's actually called Do Not Stop Me. So maybe I want to go dig into that one and see before I actually go run this.

Wouldn't it be great if life had a what-if command? Well, I guess I'll have to deal with PowerShell for now. OK. So I showed you how to manage this right here in PowerShell. Let me show you some of the really cool PowerShell -- excuse me, management experiences that Scott just talked about.

So right here in the portal experience I've got a virtual machine running. And down here in operations you'll see a few new things. Disaster recovery, update management, inventory and change tracking built right here into the portal experience. And so if I go ahead and click on update management what this is going to show me are all of the updates that have been installed or need to be installed on this virtual machine. So this is showing me OS updates right here in the portal experience, so you can see I'm missing this Windows Defender, so I may want to go install that. It shows me the updates that have succeeded and I can go into the details on exactly what those updates contained.

I could even go in and schedule the updates right here in the portal experience. And so I can say critical update, security updates, pick which one. I can exclude specific KBs right here in this portal experience. But what's great about this is it doesn't just work on one machine, it works across all the machines of my infrastructure. So if I click across manage machines across my entire infrastructure it will show me all of the updates across everything. It will show me the updates across all of my Linux machines.

It will show me updates across all my Windows machines. So you can see here I've got a few Linux machines that need updates. So these are OS updates for Linux and Windows right here in the Azure Portal. But wait, it's not just Azure. You can manage and track your updates that are required for on-premises machines, too. So right here in the Azure Portal you're doing updates across on-prem, in Azure, Windows or Linux all in the same experience. (Applause.)

That's right. That's exciting. But wait, there's more. I can also do change tracking. So here you can see I'm running a virtual machine. This is tracking every change that's happening on my virtual machine, every file that's changed, every event that happens, every registry change that happens. I can zoom down and actually take a look at all the changes and search for things that I'm looking for.

Again, not just a single machine, though. If I want to look across my entire environment I can see all the changes across all my environments, Windows or Linux, and decide and figure out what's going on, these last 24 hours. I can also zoom in, hey, you know, registry updates those make me a little bit nervous. Let me zoom into that. And let's search here and see what that update was.

And so right here, if I typed correctly, right here you can see and zoom in and see what is that change and what is it doing and go investigate if you need to. It's excellent.

OK. So that's change tracking. Now, one of the really important parts of managing your infrastructure is being able to handle disaster recovery. Let me click on this. Now more than ever, given the amount of global disasters that we're seeing just impacting so many people, making sure you have a plan for what you're going to do with your infrastructure, with your applications in the event of such an occurrence is really important. And traditionally it's really, really hard. It's really hard to set that up. You've got to make

sure you have the space, make sure you have the hardware. You've got to make sure you test it, because a disaster recovery plan is not any good if you haven't tested it.

So here now with Azure we make it incredibly easy. So I click on disaster recovery. I can configure site recovery right here, pick my target region and it will show me a picture of exactly what that will look like when I do this fail-over. So I can pick from any of the Azure regions to do this.

Now, that's not all. Once I've configured that if I want to run a test fail-over it is literally a single click. So I'm here in disaster recovery. I've got a second virtual machine. I can decide the virtual network that I want to deploy it into and hit OK. And that's going to set this virtual machine up in a test configuration in that other location for me to make sure my app and my environment works while keeping the original production one up and running, single click disaster recovery, automatic. I hear clapping over here. Thank you.

There we go. Thank you, that's awesome. Very, very cool.

Now the key point here, this isn't just a single machine. You can do this across a set of machines, build a recovery plan across many machines, do them in order and actually run scripts according to that. So I've shown you disaster recovery. I've shown you change management and I've shown you update management. The last thing I'm going to show you is monitoring.

And so with monitoring we have log analytics built right here in the portal experience. And if I click on overview it's going to give me a set of prebuilt operations that I can run, prebuilt set of stats that I can look at, things like the amount of threats that I have from antimalware and so on all built in. But if I want to do something a little bit more detailed I can click on analytics here and go straight into the analytics designer.

And this allows me to actually take a look at my queries. And you can see these queries are quite, quite simple actually to write. They're very SQL-like and they allow me to do very custom things. So in this case you see here I'm running over the last seven days. I'm going to look at the processor time of all my virtual machines in this specific subscription and it's going to group them by computer and then it's going to show you the time chart. And so let's back out here and you can see it shows me a very pretty time chart that's going to show me spikes of all those CPU processor times across those seven days.

Now let me do something a little bit deeper and let me zoom here and show you this one. This is going to look at my SharePoint farm. I'm going to run this. And what this is doing is it's actually going to look at the average CPU compared to the IIS requests on that SharePoint farm and map it out in a visual way so I can see if they're related, if there's correlation there. And it then just renders that time chart, very simple query to do incredibly custom things across millions of records, and in seconds I have this beautiful

chart that will show me, well, it looks like maybe this period there was some correlation. Maybe I should dig in and understand what's going on.

And even further, let's say I want to understand which browser I'm running, which browser is hitting my site. I can very easily do that, again, millions of records, every user agent string in my entire deployment. And I can see who is calling what from where. And you can see there's actually been a spike of Firefox usage right there.

So I showed you a gigantic virtual machine. I showed you nested virtualization. No other cloud can do nested virtualization. I showed you PowerShell built right into the portal and you can also run that on your iPhone, no other cloud is doing that. I showed you update management and change management. No other cloud is supporting that. And I showed you disaster recovery with a single click between sites. No other cloud is doing that. And finally I showed you very simple query language to go across millions of records in a second to create a beautiful picture for your analysis and, you guessed it, no other cloud is doing that.

I hope you have a wonderful Ignite.

Back to you, Scott. (Applause.)

SCOTT GUTHRIE: So Corey showed you kind of the built-in management capabilities that all of you have and can basically take advantage of out of the box when you use Azure. And all this really helps you build and really deliver hugely successful solutions on top of the cloud. HSBC is a great example of a company that's realizing tremendous success with Azure right now. HSBC is one of the 10 largest banks in the entire world and they've now deployed a number of solutions on top of Azure. Here's a video of them talking about the success they've had and the IT agility that Azure is providing them.

(Video segment.)

So it's an incredibly exciting time to be a developer right now. The opportunity to build new types of applications that can engage with end users in richer ways has never been greater, whether those solutions are native mobile apps, responsive web apps, or through new engagement experiences like bots and mixed reality. And the combination of our Visual Studio family of tools and Azure Cloud Services provide you with best-in-class support to build and run these types of experiences.

And with Visual Studio 2017 and our new .NET Core 2.0 release you can take advantage of the latest in container-based technology to deploy and scale your backend solutions. And with the Xamarin technology now built into Visual Studio you can now build rich native app experiences that work on any platforms, including iOS, Android and Windows-based clients. And do so in an incredibly productive way that enables you to do massive code reuse across all the different native mobile app experiences.

What I'd like to do now is invite James on stage to show you all how you can do this.

JAMES MONTEMAGNO: Let's do this, yes, all right. I am absolutely delighted to be here with everyone to show you how we're making it drop-dead simple to build out beautiful native mobile applications with Xamarin and cloud-connected experiences with ASP.net core. And where you get started is where I always get started, right inside of Visual Studio.

I'm right inside of Visual Studio and all I have to do is say file, new project. Inside of the new project dialogue we have lovely templates for Android, iOS, Apple TV and cross platform templates to build your mobile apps and your ASP.net Core backends.

Now I've started to build an application here. I want to focus on mobile first. I'm building an expense application so I can send Scott all my expenses after Ignite. So over here we have a Xamarin solution with my actual ASP.net Core Web API backend. And here I have things like models and cloud service backends, view models and actual XAML views. So I'm sharing 100 percent of my source code and user interface between iOS, Android and Windows. And I still have those projects here.

So if I want to actually write the platform-specific code I have access to 100 percent of the APIs all in C#, which is delightful. Now, to get started, though, we know that there's complications with emulators, simulators, max build machines, all this stuff. We wanted to make it simple so anyone could get started with just an iOS or Android device and their PC with Visual Studio.

So let's get my iPhone up here. I have a real iPhone and what I'm going to do here is launch my actual brand-new Xamarin Live Player application. What I'm going to do is over the local Wi-Fi is pair Visual Studio to the Xamarin Live Player device. So let's go ahead and pair it here. And I'm going to scan it. There we go, and boom right over the Wi-Fi and connected to Visual Studio and ready to deploy my applications.

So here I have my Azure iPhone Player, which is this iPhone right there. I'm going to hit debug just like I always have. But over the local Wi-Fi connection it's going to send all the files to my iOS device and it shows up in mere moments, being interpreted on the device. That's crazy. It's ridiculous. It blows my mind every time I do that. But it's a real app. So I can browse through. I can actually hit add here. And I'm actually inside of a real live debug session with a breakpoint. So I can browse through all the code here. I can continue on and use the application as I would expect.

So let's go browse a little bit and I need to spice up my wardrobe with some red polos to match Scott. So those are, what, like \$9.99. And we'll add some geo location maybe. And, heck, let's just add a receipt so I can send this off to Scott. So I'm using my application and I'm using cross-platform code inside of this project to access native features like geolocation and to get photos. So we can continue on.

But what I love here is that we've made it really possible to not only debug over local Wi-Fi to a physical device, but to also rapidly iterate on the user interface. So I'm going to

go back to this list page of all of my data. I'm going to say Xamarin Live Player live run this current view. And immediately my application shows up. And I can just start working on it.

So here what I'm going to do is just go over here and I have a header I want to add. So I'm going to drag and drop it in. Visual Studio waits for me to basically stop editing and, boom, it updates in real time. I can come back over and if I want to make more code modifications, so let's say red, I can go ahead and do that, it shows up. And as it waits for me to finish redeploys and updates again right over the local Wi-Fi without having me even save. (Applause.)

Now let's switch over to my ASP.net Core backend and get rid of that iPhone. So inside of this project I also mentioned this application is reaching out to my ASP.net Core Web API. So inside of this project I have a .NET standard library where I'm sharing code between them, as well. Now I've been debugging it locally on my machine here at Ignite. But I want to make sure that when I deploy it it can work absolutely anywhere. And what I'm going to do is leverage Docker.

Now I don't know much about images and containers. Donovan tries to explain it to me every single day. I still don't get it. But all I know is I need to add Docker. So what do I do? Let Visual Studio handle it by right-clicking and saying add Docker support, one click and it will automatically scaffold out all the infrastructure I need to put this into a Docker image.

So here is essentially the Outline for this application and Visual Studio not only adds the Docker file, but it adds Docker Compose, so I can deploy it later. And here Visual Studio knows that, hey, this is now Dockerized. I can just hit debug. Visual Studio leaps into action and packages up this image and then runs it inside of Docker running on my machine. There it is. Now I'm spinning up my ASP.net Core web application and then it will launch the browser into a real live debug session.

So here's my Swagger definition. Here's all the items that the app was reading out, say try it out and I'm inside of Visual Studio debugging a Docker image from Visual Studio and Docker, just like I always have, awesome. (Applause.)

Now, when I'm ready to deploy, though, I want to make sure I can take it absolutely everywhere and anywhere. So I put in a Docker image. So, of course, I'm just going to let Visual Studio handle all of the hard work when I want to publish. What do I always do? Right-click, publish, of course I do. That's how I get it up into Azure.

So what's nice here is that since I've added the Docker file and Docker compose support, it knows that I can then take that and deploy that Docker image into Azure Container Registry, or better yet, since it's a Web API, it can do that and then also deploy it immediately out into an Azure App service. So I hit publish, now Visual Studio knows that this is a Docker image that it needs to deploy the container registry and it handles all of the heavy lifting for you with a single click, awesome.

And I've just shown you now how to build -- yeah, you can clap, it's awesome. (Applause.) I've shown you how to get started with mobile development with nothing more than your iOS and Android device with Visual Studio and how to build beautiful web backends with ASP.net Core and to publish it up into Docker images all from Visual Studio.

Thank you so much and have a great Ignite. (Applause.)

SCOTT GUTHRIE: UPS is one of the largest shipping and logistics companies in the world and they are now using the combination of .NET Core, Xamarin and Azure to build rich, new customer and employee experiences. Here's a video of them talking about the success they've had.

(Video segment.) (Cheers, applause.)

So the cloud doesn't just bring new technologies and features. It also introduces new approaches that you can use to accelerate the delivery and the continuous improvement of your solution. DevOps is an approach that all of you have heard about, and I talk a lot to customers around the world who are looking for ways to adopt this within their existing organizations. But setting up a DevOps pipeline, one that spans your build, detect and deployment environments that has integrated analytics throughout it and which is optimized for a team environment, it's traditionally not been easy. And it's required you to manually integrate a lot of different tools and systems together to get a solution.

With Visual Studio Team Services, we're making it much easier to adopt a DevOps-based model and set your team up for success. VSTS is fully integrated into Azure and includes everything you need and works with every language and run-time environment you already have.

What I would like to do now is invite Donovan Brown on stage to show up DevOps in Azure using VSTS. Here's Donovan.

(Applause.)

DONOVAN BROWN: Thank you so much, Scott, and good morning, everyone. Let's try that again, good morning, everyone. (Audience response.) Perfect.

Today we are all honorary members of the Visual Studio Team Services team. And with that honor comes the responsibility to ship every three weeks. To ship at that velocity and keep our quality, we have to employ DevOps and best practices. So it's Monday morning, we're walking into our team room, and on our Surface Hub is this dashboard. This dashboard is the game plan for our sprint. It shows me everything assigned to me, what pull requests I need to review, what bugs I need to resolve. It even shows me how many days are left on our sprint so that we can focus and deliver on our goal.

Is anyone old enough to remember when your bugs were in one system and your user stories were in a different system and your tasks were in yet a different system? What am I the oldest person in here? Come on. Those days, luckily are gone. Now I have a product backlog that shows me my user stories, my bugs and my tasks all in one place and in priority order so that we can focus on the most important things first.

Now if you're anything like me, you hate writing status reports. Now, I get it, I know why they're important, but I always feel that that time is better spent adding value to my product. So I love our new Kanban board. This, this is a real-time status report. At a glance I can see what all my team members are doing. I can tell if there's an item that needs my attention. And to update my status I simply drag and drop from one column to the next.

Now on our team we use Git. And if you know anything about Git, that means branches and lots of them. I used to create so many branches I would forget why I created the branch. But luckily if I use the Kanban board to come and create my branch what's going to happen here is that this branch is now going to be associated with that work item. It doesn't end there. Every commit that I make to that branch will also be associated to that work item. But, wait, if you order now, I'll add your build and your releases as well, everything automatically gets tied together for you. I can open up that work item, show you every line of code that was changed, every build that was triggered and what environment it is currently running in all from creating a branch.

Thank you, you can clap for that. That's awesomeness. (Applause.)

Now eventually all these branches have to come back together. We do that through something called a pull request. Now I've been writing software for over 20 years. I write software because it's fun. And I want a pull request to also be fun. So what we're going to do is, we're going to bring a social aspect to pull requests. I can open up a pull request that's assigned to me, and I can now communicate with my peers as if I were on any other social network. If the best way to communicate is through a meme or an emoji, I can do that. And if some code just gives you that warm and fuzzy feeling, I can actually share that with my team right here in my pull request. Oh, that feels so good, right? We've all seen that line of code that just gets us excited, and now I can share it in my PR.

Now that the code has been pulled back into master, it's time for us to deploy that code into production. Now it's hard to do a show of hands when you have 26,000 people, but is there anyone out there that still manually deploys your software? No one? Give me a break. But gone are the days where we throw our code over the fence like a hot potato. Now we pass it from team to team like a baton. This is where we break down the walls between dev and op.

What we're going to do is we're going to bring the power of Visual Studio Team Services right in the hands of the IT pro in the Azure portal. Now we're letting the IT pro rub a little DevOps on it and make it better. We're going to basically give you self-service DevOps. I simply click on the tile that represents my web app, and now I'm presented a

blade that allows me to set up continuous delivery. I get to use the power of VSTS, but I get to use it from Azure. As an IT pro, I no longer have to wait on my dev team to setup CI and setup CD. I can do it now myself. I can be the change agent. I can be the one forcing the DevOps best practices upon my organization.

Step one, choose where I want my code. I can use Visual Studio Team Services or I can even use GitHub if I'm working on open source. You do not have to move your code to take advantage of VSTS in Azure. I'm going to use VSTS. I'll choose the account because it knows who I am. It's going to load my project, and it's even going to load the repository and the appropriate branch for me to make sure that I can actually go ahead and configure this. So there's my branch being loaded, and now I'm simply going to click on OK.

Once this OK is enabled what I'm going to do now is I'm going to choose what language I want to program in, because you've got to remember this, this is not your daddy's Microsoft. This is a Microsoft that understands Node.js, Python and PHP. We could run this on Linux and we can run this on Windows. It's about time we tell the world, if you write software, here at Microsoft we add value for any language targeting any platform. (Applause.)

Now all I have to do now is click on OK. This would be a great time to take a picture and send a tweet, but only send one because this will be done in a minute. What we're doing in the back end is Visual Studio Team Services and Azure are working together to build an entire CI, CD pipeline that's going to allow the code to flow from the fingertips of my developers into the hands of my users.

As the IT pro you've always been the protector of our infrastructure. But now you can be the guardian of our pipeline to make sure that code continuously flows from our fingertips to our end users. This is being built for you, but you are in complete control. Now that it's done I can take you into the build that was created for you. This build definition that was created for you is under your control. You can add to it whatever you need to to be successful. As you can see, it's already running in resources that were waiting for you in Azure. We have Linux build machines and Windows build machines just waiting to build your code for you.

While that build is running, I want to quickly show you the anatomy. It's just a series of tabs, all the tabs that are open source that you can go see exactly how we created them. You can create your own. You can add additional ones, or, better yet, you can actually go to our marketplace, here we have new extensions added every day that add value to your CI and CD pipeline. We have to remember that DevOps is the union of people, process and products to enable continuous delivery of value to our end users.

With Azure and Visual Studio Team Services we can now do that for any language targeting any platform. Thank you so much, everyone.

(Applause.)

SCOTT GUTHRIE: So we've looked at how productive you can be when you're building solutions using Azure. And one of the great things about Azure, of course, is the fact that it gives you this productive experience everywhere. One of the biggest differentiators with Azure versus other cloud providers is our hybrid cloud capabilities. Microsoft is the only technology company delivering a complete hybrid cloud. And this is important because hybrid cloud isn't just a trend or a buzzword, it's the reality that every organization has.

Hybrid does not mean just connectivity between your on-premises data centers and a public cloud. Real hybrid means integration and consistency across your entire technology estate, from your identity system to infrastructure and developer platform to your data estate through to your management and security operations.

And one of the great capabilities that we're really excited about that's shipping today is Azure Stack, which is a key part of this story. Azure Stack is an extension of Azure. Azure Stack provides a consistent cloud experience with the same management API, the same set of developer services, same management portal that's available in Azure. And it's one that you can deploy and run literally anywhere.

For example, with Azure Stack you can now use the cloud for edge and disconnected scenarios, like on a ship out at sea where the network might not be reliable, or a factory floor where you might need incredibly low latency to your embedded automation systems for production. The combination of Azure and Azure Stack also now allows you to meet literally every regulatory need. Azure today has more compliance certifications than any other cloud provider. It runs in more countries and regions than any other cloud provider, as well.

Yet, we know there's still going to be scenarios where the data and applications need to reside in a specific country that doesn't yet have a public cloud provider with a data center. And this is where Azure Stack now allows you to complete that story. Let me show you an example of one company that's leveraging this. EY is one of the largest professional services firms in the world and they're using Azure to run many of their applications. And being a trusted and secure cloud provider was the key reason why they chose Azure. The documents and data that they store on it are amongst the most sensitive pieces of data in the world.

And the challenge they've had is being able to use a single cloud application for all their clients. Some countries in the world, for example, Russia, require that company documents can't leave their soil and no public cloud provider currently operates in that country. And the beauty of Azure Stack is that it gives EY now the ability to write their application once and deploy it both in Azure and run it in their own facilities in those countries with Azure Stack. This gives them the ability to meet literally every regulatory requirement in every country in the world with a single code base.

Let's watch a video of them telling their story.

(Video segment.)

(Applause.)

SCOTT GUTHRIE: And Azure Stack also enables you to begin modernizing your on-premises applications even before you move them to public cloud. And the great thing is it's super easy to use, and it's 100 percent consistent with full Azure.

So what I'd like to do is invite Natalia onstage to show off Azure Stack in action. Here's Natalia.

(Applause.)

NATALIA MACKEVICIUS: Thanks, Scott.

I am really excited to be here, but even more excited that we are shipping Azure Stack Integrated System.

So let's take a look at the edge and disconnected scenario that Scott talked about, and how Azure and Azure Stack working together can address that edge and disconnected scenario.

So here I have a shipping company, Northwind Traders, that has a global fleet-wide application. They also have local processing that needs to be done on each ship, and they're using Azure Stack for each ship. In the global application you can see each ship and their location in the waters.

If I go to the Azure portal, you can see all of the application characteristics here that build the global fleet-wide application. We have Event Hub. Event Hub is being used to receive and process massive data streams from each ship. We also have Web App that's being used for that live viewer monitoring on the ships.

So let's switch over to the systems running on the ships. We're going to switch to Azure Stack. Now, if you look at the portal here, this is Microsoft Azure Stack. The portal looks identical. That's the entire point, there is API consistency, so the portal will be exactly the same in Azure as it is in Azure Stack. You can leverage people skills, processes and applications across both Azure and Azure Stack. You can also have IaaS and PaaS services.

Just like we had Web Apps running in Azure, we have Web Apps running in Azure Stack.

Here we have Functions. Remember, this is a disconnected scenario. You're in the middle of the ocean on a ship. And on that ship bandwidth is super expensive. So it's a

perfect use case for Functions to parse the data. You can decide what flows to Azure to the global system.

We can monitor all the events and have consistency from Azure and Azure Stack. It's the same server-less computing.

Let's take a look at how this hybrid application was developed.

I'm going to switch over to Visual Studio, and here we can see the application components for that global application running in Azure.

You can also see the application components running in Azure Stack. Here we can see Functions that we talked about earlier, that server-less computing, being used to parse data.

We can make changes to an application. Now, I know Donovan said we have Visual Studio Team Services, but here I'm going to view a manual deployment. I could use CI/CD Pipeline to actually deploy the application, but I'm going to manually kick off the deployment of the application to Azure Stack.

Because I'm using Azure Active Directory, I can choose either Microsoft Azure subscriptions or Azure Stack. So here I'm deploying to the local Azure Stack implementation on the ship. If I kick off the deployment, then I can see here is that application running on Azure Stack.

That's the power of Azure and Azure Stack. Azure Stack truly is an extension of Azure. It gives you the capability to use the same processes to build, to deploy and to operate your applications wherever your business needs.

Thank you. Back to Scott.

(Applause.)

SCOTT GUTHRIE: I'm incredibly excited to announce that as of today we're now shipping Azure Stack Systems.

(Applause.)

We now have certified Azure Stack appliances made by HP, Dell, Lenovo and Cisco. They're on display at the expo floor here at Ignite, and they're now all available for order. And we're really excited to see the great hybrid cloud solutions that you're going to stand up on top of them.

So with Azure Stack you now have the ability to stand up a local Azure cloud anywhere in a matter of hours. No other cloud gives you this type of hybrid consistency and flexibility.

Now, one of the most complex, as well as the highest value aspect of any application is how it actually uses data. And dealing with data in a hybrid application or a full cloud migration situation has historically been very challenging. And one of the great things about Azure is how it enables you to use SQL Server consistently across your entire hybrid cloud estate.

Earlier this year, we introduced our new SQL Server 2017 release. SQL Server 2017 is the fastest, most secure and most intelligent database on the planet, and now provides ultimate flexibility.

We're releasing SQL Server 2017 simultaneously on Windows Server, Linux, as well as Docker-based systems. In fact, we have more than 2 million pull requests of our SQL Server Linux image from Docker Hub over the last three months alone. (Applause.) There's one of them.

SQL Server 2017 also delivers unparalleled performance with our new adaptive query processing engine, making it the fastest release of SQL Server ever.

And we now enable you to use machine learning models built with R and Python, and run them in memory directly inside the database, making your apps more intelligent than ever before as well.

And really the combination of this flexibility, performance and built-in AI is winning new fans to SQL Server, including from startups and enterprises that frankly never considered using SQL Server before.

Let's hear from DV01, a financial services startup, about their experience and why they decided to switch to SQL Server 2017 on Linux. Here's their story.

(Video segment.)

(Applause.)

SCOTT GUTHRIE: So let's now take a look at how developers, including those not using Windows, can install and start using SQL Server 2017 running in a Docker container today.

Here's Lara to show it off.

LARA RUBBELKE: Thanks, Scott.

Hello, Ignite! How exciting. It's fantastic to be here and show off SQL Server 2017. We've got a great release, and I'm super thrilled to just be able to show off what we've got going on.

As Scott mentioned, we can run this and deploy this anywhere. That includes Windows and Linux. And that's great news, because I'm running on a Mac right now, and I don't have SQL installed yet.

So the first thing we do before we start getting into some of the advanced features, I need to install SQL Server.

So to get started, I'm going to pull the latest SQL image from the depot. And once we have that image downloaded, let's light up our image.

As I light up my image, I'm going to give it my password, I'm going to give it the port numbers, and that's it.

So from here we're going to do our installation of SQL Server, if you bear with me for an hour or two while I install SQL Server, and we're done. How about that? (Applause.)

So it appears like we've got SQL installed. Let's just make sure that SQL is running. And to do that I'm going to connect to SQL Server using our tools, the SQL Server tools installed right here on my Mac. And we'll just run a quick statement. And everything is running, so we're all set to go.

All right, let's end process.

OK, now that we know that we've got SQL Server running, I'd also like to test to make sure I can connect to this with external applications.

SQL supports all the major programming languages, PHP, Python, Ruby, all the major languages. And in fact, I've got an application, a very simple application already set up here, and this is a Node.js application that's using the open source TDS drivers. And all it's going to do is just issue a connected once it's able to reach SQL Server.

So we'll come out back to our terminal and let's issue our application, and we're connected.

But how about this? In just about a minute I downloaded, I installed, and I connected to SQL Server. Pretty impressive, isn't it? (Applause.) All right.

Now, this is not just a database engine. This is an engineering-grade database platform, with all the advanced features for maximum security, for robust performance, for manageability, everything you expect out of this engine.

To give you an example of how you can use this for your advanced workloads, let's have a look at how we can improve application performance with SQL Server 2017.

For this I need data. I also need a map. So we're just going to start with data. Now, I'm going to issue this command, and this command will create an order table. Come back out here.

This order table will have a couple of different interesting columns, and it's going to end up with 5 million rows of data, just enough data to be interesting.

For an app I have an ASP.NET core application, and it's connecting to my local instance. As you can see, it's just going to start by issuing a select sum over the price column of that orders table.

So let's go ahead and run our app. We've got everything running. Let's connect.

Now, what I'm doing is issuing just a baseline. We want to find out how the performance is with our SQL Server before we make any changes. So we're starting at 231 milliseconds. I know I can improve performance with SQL Server 2017. So let's go back and make one minor adjustment. And what I'd like to do is create a clustered columnstore index. So we'll connect back out to SQL Server.

A clustered columnstore index is unique to SQL Server, nobody else has this. It stores the data in a special order on disk. This maximizes decompression, and it makes performance just super, super fast. And that's great for things like aggregate queries like we had on our query earlier, our sum.

So if we come back, let's rerun our application and see how our performance improves. We're down to six milliseconds. (Applause.) That's from a single line of T-SQL, one line.

Now, what's really fantastic is that in the past features like clustered columnstore index, as well as in-memory tables or row level security or compression, as well as many others, were only available in the enterprise edition of SQL Server. Well, today, they're available in every edition of SQL Server, including SQL Server Express.

(Applause.)

So go download SQL Server 2017 today, start building some apps, and have fun with your new database engine. Thank you.

(Applause.)

SCOTT GUTHRIE: So I'm incredibly excited to announce the general availability of SQL Server 2017 on Windows, Linux, and in Docker containers. You can download the release and start using it today.

(Applause.)

Now, one of the things that makes SQL Server unique is that it's really the only data platform out there today that's available both on-premises and as a fully managed database service in the cloud using Azure.

What this means is that you can now take advantage of all of that SQL Server functionality, including columnstore, in-memory, all the great stuff that Lara talked about, anywhere, and get the best possible experience now when using it as a fully managed service in Azure.

You can stand up a SQL database in Azure in under 60 seconds, and have it be highly available, durable, secure and fault tolerant, without you having to configure anything.

Azure provides built-in backup and point in time restore capabilities, automatic performance tuning support, and threat detection and security capabilities that enable you to securely run your system.

And best of all, you get all these capabilities without having to manage virtual machines or worry about patching or tuning your infrastructure manually. Our SQL database as a service offering takes care of all that for you.

We now have a new Azure data migration service that will make it even easier to migrate your existing databases to use our SQL database as a service in Azure.

The data migration service streamlines moving existing database systems to Azure, and provides a fully automated workflow to do so, both for on-premise SQL Server databases, as well as for other non-Microsoft database platforms, including Oracle database systems as well.

And the new data migration service, combined with the capabilities that we're adding to our SQL database offering this year, is going to make it trivially easy to migrate any existing SQL Server database to the cloud, without having to change any code in the application, and have a seamless, near zero downtime migration experience.

Now, DocuSign is an example of one Azure customer taking advantage of this new capability. DocuSign is one of the leading enterprise SaaS providers today, supporting 200 million different users. And they've historically run their own transaction processing database systems within their own data centers. And we're honored that they've chosen Azure as their preferred cloud platform to run their systems going forward.

Here's a video of them talking about this decision.

(Video segment.)

(Applause.)

SCOTT GUTHRIE: And what I'd like to do is invite Lara back onstage to walk through what it looks like to migrate a complex SQL Server based application that exists today to Azure with zero downtime and no code changes required. Here's Lara.

LARA RUBBELKE: Thanks again, Scott. It's great to be back.

Many of you have applications that are powered by SQL Server. And many of you are migrating and modernizing with the cloud.

Now, there are a class of applications that have complex database requirements. They require things SQL agents, CLR, the cross-database joins or VNET. And in the past, these dependencies prevented you from taking advantage of the modern PaaS database experience.

Now, the great news is that all of these and more are now supported with the new Azure SQL DB managed instance.

Let me show you how this works. I have Stack Overflow's enterprise application deployed locally. And if you scroll down to the bottom, you'll see that we are connected. Look, we're connected to SQL Server 2008 R2, hosted in my data center. This application is dependent on two databases, and requires cross-database joins to render these views.

Let's migrate these databases to Azure. To migrate to our new Azure SQL DB managed instance I'm going to use the database migration service. The service is very intuitive, super easy to use.

I start by giving it the name of my target. Sorry, I guess I already clicked and we're running it. So we give it the name of the target, and then we add in the name of the source, SQL Server 2008 R2 environment. This can be either any supported version of SQL Server or it could also be Oracle or MySQL

With a couple of additional settings we just set run migration. And as you can see, we're already running our migration.

If we click into the progress, you can see that our databases are already restoring, and in fact, one of them has completed.

Now, this supports multiple scale types, everything from small megabyte-sized databases up to multi-terabyte sized databases, all with minimal downtime.

Now let's just click refresh, if we come back to our service and we refresh up here, you can see that we've already succeeded. So we're migrated and we're now running in the cloud.

Now, I'm going to come back to my application, and in my application we're going to change our connection string, and now I'm going to point it to that managed Azure SQL DB managed instance that I created earlier this morning that we migrated our application databases to.

Now, if I run my application locally again, it's going to connect up to our managed instance, and there you are. We have our cross-database joins rendering correctly, without any changes to my application. If we scroll down to the bottom, you can see that we are connected to Azure SQL DB managed instance.

(Applause.)

Thank you.

Now, this is not a virtual machine, this is a managed PaaS database service. I don't have to worry about updating my operating system. I don't have to worry about upgrading SQL Server. I don't have to worry about managing machine configuration. Azure takes care of all of this and more for me.

And I get the advantages of running SQL in Azure. Azure is the best cloud for SQL Server. It will help me maintain the highest availability, the strongest security, and the maximum performance for my application databases.

That's it. I appreciate your time. Thank you.

(Applause.)

SCOTT GUTHRIE: You saw how easy it was for Lara to migrate a large, real world application that featured StackOverflow.com, from a SQL Server running on-premises to run using our SQL database as a service offering inside Azure, literally with no code changes.

And the ability to easily migrate workflows also leads to incredible cost savings with Azure versus other cloud providers.

The data on this slide here is from one large enterprise customer who evaluated a cloud migration to both AWS and Azure for literally hundreds of their different Windows Server and SQL Server application workloads. And the ability to easily migrate their apps without having to make code changes, combined with the ability for them to easily use their existing Windows Server and SQL Server licenses they'd already purchased, enabled the migration to Azure to literally be 70 percent more cost effective, in this case here enabling them to save millions of dollars on top of that migration.

Now, data is fundamental to developing breakthrough intelligent apps, and Azure provides a complete set of data and AI services that enable you to build these types of

transformational solutions, and to do them at planet scale, and to enable your businesses to reach literally every customer in the world.

Earlier this year, we released Azure Cosmos DB, which is the first globally distributed, multimodal database service that delivers horizontal scale-out with guaranteed single digit millisecond latency. It's designed to explicitly handle these types of next generation intelligent apps.

You know, imagine a horizontally scalable database that puts data everywhere that your users are. With Cosmos DB we've built a database service that does just that, and can automatically replicate your data to any Azure region around the world, to give your users lightning fast performance regardless of where they're accessing your application.

Cosmos DB also enables you to elastically scale your storage and performance throughput across one or multiple Azure regions, with zero application downtime. You can start with say just gigabytes of data, and then scale to manage petabytes of it. You can start with processing just say 100 operations per second, but then scale up to handle tens of millions of operations per second.

And best of all, with Cosmos DB you pay just for the storage and performance you need, enabling you to save a tremendous amount of money.

Cosmos DB starting today also now allows you to run server-less code in response to data changes inside your Cosmos DB database.

Using our Azure Functions service, which enables you to dynamically run code without having to have a dedicated VM to provision, you can now run code at a global scale, and pay only for the compute cycles you need, all integrated with the data that you store inside Cosmos DB. This provides a fantastic way that you can infuse intelligence logic even deeper with your data.

And Cosmos DB is also the only database with comprehensive SLAs across availability, data consistency, and performance. In fact, Cosmos DB, as I mentioned, guarantees single digit millisecond response time at the 99th percentile as one of the SLAs. And you can directly monitor this and all the other SLAs directly inside the Azure management portal.

ASOS is a great customer of ours running in Azure today, and it's really a fantastic example of the type of applications that can really take advantage of a database service like Cosmos DB. Here's a video of them talking about ASOS.com, and how they're leveraging Azure.

(Video segment.)

(Applause.)

SCOTT GUTHRIE: Now, not only is Cosmos DB incredibly powerful but it's also really approachable from a developer perspective. And what I'd like to do is invite Rimma, one of the architects of Cosmos DB, onstage to show off how to build solutions with it. Here's Rimma.

(Applause.)

RIMMA NEHME: Thank you, Todd. Thank you.

Good morning, everyone. It's really exciting to be here to show you Azure Cosmos DB, our globally distributed, multimodal database service.

Now, imagine you want to build a planet scale intelligent app. It has to be global, it has to be super-fast, it has to be incredibly elastic and high available across the globe. In this demo I will show you how easy it is to do it with Azure Cosmos DB.

We've taken a Marvel universe dataset, which represents a complex network of comic characters, and built a simple web app around it, where fans from all over the world can come in, pick their favorite character, and then ask any question that they want, for instance, Iron Man, who are your friends, and get the response back.

Now, behind the scenes we've taken this dataset and stored it inside Azure Cosmos DB as a globally distributed graph. And all of the questions that I'm asking get translated as graph queries and get processed by Cosmos DB in near real time.

So how do you build such an app? In fact, how do you even create a globally distributed database that spans the entire planet? It's incredibly easy. Let me show you how.

You come to Azure portal, click new databases. Then you pick Azure Cosmos DB. And then you specify all the normal parameters you normally would for any other service like name, subscription, resource group.

But the most interesting here to pay attention to is this set of APIs that you can actually pick from. For instance, if you are coming from the relational world and you want to pick a SQL API for interacting with your data, you can pick SQL APIs. If you're coming from the MongoDB world, you can pick MongoDB APIs. If you're working with graph data, you can pick a Gremlin API. Or for key value storage you can pick a table API.

So you pick the API of your choice. Then you can go and specify other parameters. You pick the location of your database where you want to start out, and then you just click create button, and that's it, you're done. This is how you create a globally distributed database with very powerful capabilities.

Now, our dataset is already stored inside Azure Cosmos DB, and I can come to the portal, click on the data explorer, and actually visualize it as a graph. I can click on the nodes. I

can see the connections between the nodes. I can actually specify a Gremlin query directly inside the Azure portal and then get the result back in the form of a subgraph.

Now, remember this is not just a regular graph, this is a globally distributed graph. In fact, it is already distributed across ten regions worldwide.

Now, let's assume my application goes viral today after this demo, and I have fans now in Southeast Asia who want to come and interact with my app. All I need to do is just select these regions, and then click the save button, and that's it. The data gets seamlessly replicated into those regions, and becomes almost instantaneously available for querying in those regions. This is what we call a global distribution turnkey capability.

And as I'm adding and removing regions, my application continues to be highly available throughout all the time, thanks to the multi-homing capabilities that Azure Cosmos DB provides.

Now, the key thing to remember here is that regardless of the data model, regardless of the API that you choose, regardless of the data distribution that you pick, regardless of the failures that may even occur at such a large scale, we continue to provide you with a single system image for all of these globally distributed resources, and we continue to give you horizontal scale-out of both storage and throughput, and single digit low latency response rate, without you having to actually provide any indices.

So finally, the SLAs. Now, a true measure of any cloud service is measured by the SLA. And Cosmos DB is the first service in the cloud industry that gives you comprehensive SLAs for latency, throughput, consistency, and high availability at the 99th percentile worldwide, so you can put your mission critical applications on Cosmos DB and run them with a peace of mind.

Now, today we are really excited to announce something super exciting, native integration between Azure Functions and Azure Cosmos DB. With this native integration you can easily create event driven, server-less apps, with low latency access to diverse data, all across the entire globe. You can create a Cosmos DB trigger, and then have Azure Functions respond to the events triggered directly by your data, and run the code that you specified without you having to worry about scaling or managing computing resources.

Now, we've added a capability to our app to monitor all of the social channels like Twitter, Instagram, Facebook, for all of the events that are happening across the globe that are related to Marvel characters. All of these updates arrive to Cosmos DB in the form of the updated event, and then we have an Azure Function that gets triggered upon the arrival of those events, and then calls the Cognitive Services API. In this case it happens to be a text analysis API that goes and parses that event and then generates a sentiment score with respect to that event.

So now the fans can not only find out about all of the real-time events happening all across the globe, but also find out what is the emotional sentiment involved in those events.

So let me actually simulate create that. So here we've simulated a Twitter feed, and then I click save. I can come now to the app. And there you go. So this is the tweet that just exposed the event, with a 99 percent positive sentiment.

So the beauty of this architecture is that it lets you break down your app into smaller and simpler pieces that are easier to manage. It takes away the pain associated with running the servers and maintaining them to a point where you can simply concentrate on your app and the logic behind it, instead of the infrastructure.

Such architecture is also much more economical to run, because remember, the Azure Functions get triggered only when the event happens, in this case whenever a data update happens, regardless of wherever it is in the world. And Azure Cosmos DB makes your data available, highly available, close to where users are, all across the globe.

So there you have it, two very, very powerful services. Combined together, they can really transform your applications and your businesses.

Thank you so much. With that, back to you. Scott.

(Applause.)

SCOTT GUTHRIE: Now, in Rimma's application she showed how you can use Azure Functions to run server-less code in response to data changes, and she showed how you can call existing AI algorithms, in this case your Azure Cognitive Services that Microsoft's built, as part of that to make your data even more intelligent.

What I want to show now is how you can not just use existing algorithms from others, but how you can actually build your own AI algorithms with our new Azure Machine Learning service, which we're releasing a tremendous amount of new capabilities in today.

We also today decided to release the new public preview of our new Azure Machine Learning Workbench, which is a client side tool that runs on both Mac and Windows systems, and it allows you to be even more productive as you build your AI applications, and seamlessly integrate with our Azure Machine Learning service running in the cloud.

Now, what does this give you? Well, so data scientists building AI algorithms today spend on average about 80 percent of their doing what we call data preparation steps, working with data from multiple sources, merging it together, finding and fixing data anomalies, and standardizing formats. It's an arduous task that a lot of people often refer to as data wrangling.

Our new machine learning Workbench makes these types of data wrangling and modeling tasks lightning fast compared to what you've had to do before, enabling you to spend much more time building your AI models and running experiments against them.

Our Azure Machine Learning service and Workbench also provide an incredibly agile environment in which you can build and test these models. We have deep Jupyter notebook integration, enabling you to easily collaborate with others as you work on your AI models. We also have rich support for every data science and AI framework out there, including things like Spark, TensorFlow, Cognitive Toolkit, Caffe2, and more.

And once you build an AI model with Azure Machine Learning, you can now package it up starting today into a Docker container, and deploy it to run anywhere. This is a big difference from what other cloud providers offer. With our Azure Machine Learning service you can actually build now and deploy AI models not just to Azure but to other cloud providers including AWS and GCP, as well as existing on-premise environments, and even run your AI models directly on edge devices. This gives you maximum flexibility to use AI literally everywhere.

What I'd like to do is invite Danielle onstage -- she's a member of the Azure Machine Learning service team -- to show off some of these great new capabilities, and how you can take advantage of it. Here's Danielle.

(Applause.)

DANIELLE DEAN: Awesome. Hello, everyone. Thank you, Scott.

So Jabil one of the world's largest manufacturers, and they are creating an intelligent factory. I'm excited to be here today to tell you how they are embedding artificial intelligence within their circuit board assembly process.

Now, circuit boards are typically created through an assembly line, where the same board type is manufactured many times. An assembly line starts with an empty board, and through a series of operations components are mounted onto that board.

Now, during this process pictures are taken of the components to help spot errors. Human operators have only a couple of seconds to review a set of images that are flagged by the machines, and decide whether a board needs to be reworked. We want to help those operators be more efficient to focus on only the most critical images.

So let's look at some example images on my demo computer in more detail.

Now, you can see here in this picture the component has been mounted correctly onto this board, so this one is a pass. If we look at another example, the component that's supposed to be there is actually missing, so this one is actually a failure.

Now, there's lots of different types of failures. It requires experience to understand which ones are actually pass or fail. You can imagine it would be extremely difficult to code a software program to distinguish between passes and failures in these images.

But with AI it's possible to do this accurately, and to deploy a system that automatically classifies images of circuit boards into pass and fail. In turn, this greatly speeds up the efficiency of the operators in their validation process, and improves quality of the manufacturing system.

So let's see how this was created.

Jabil has given us lots of historical data. This includes the board, as well as the image pass. This is the data that we're actually going to learn from.

Now, first, as a data scientist myself I need to do some data preparation to clean up and structure this data. This usually takes a lot of time to do. So to do this I'm going to use the new Azure Machine Learning Workbench. So let's open that now.

Azure Machine Learning Workbench or what I'm going to call AML is a cross-platform desktop application that makes data scientists more productive throughout the entire data preparation and modeling process to get from data to an AI application in the shortest possible time. It's a combination of Azure services and tools for machine learning, with a development GUI, as well as command line support.

I'm going to use this tool for data preparation and then to build the machine learning model, and then to deploy it so that we can actually use it in real time.

For the first step let's open up an example project. And I'm going to do that data preparation I was talking about earlier. In this case I actually need to pull out the historical outcomes of those circuit boards analysis to understand whether it was a pass or failure in the past. And this is actually stored in the image pass.

So Azure Machine Learning Workbench makes this really easy to do with built-in data preparation capabilities. I'm going to give it a few examples of what I want, and it's actually also going to deal with edge cases. So, for example, there might be parentheses around a number, there might be different ways of storing this information. And if it doesn't get it right, I can actually give it more examples and it will improve.

Now, this is a relatively simple example. So you might be saying, why should I care? But these tests are actually recorded on the right-hand side, and the amazing part is that this is then synthesized into a program that I can then apply at scale on the cloud. Now, that is really powerful.

Now, once I've done my data preparation, I actually want to build that machine learning model. So I'm going to use those steps I did earlier, and then build the model.

Here I have a Jupyter notebook open. Jupyter notebooks are a really popular tool for data scientists to build machine learning environments. I'm leveraging a lot of open source Python packages here. They've been directly embedded within AML Workbench so that I can develop my code iteratively, and easily share my results and collaborate with others.

So I'm actually going to click shift-enter and run the code here.

Now, in this example I'm actually pulling out the data from the data preparation I did earlier. So I'm going to be then applying and creating a deep learning model on Jabil's real data.

So let's run that here, and we can actually look at an example image, and actually apply that and see what it predicts.

Now, I can change this and actually look at different images in real time, and see that this one is actually a failure.

Now, there's a small little detail that you probably didn't notice. On the upper right-hand side you'll notice that it actually says GPU Linux up here. I'm actually running this code within AML Workbench against the power of an Azure Linux data science virtual machine, with remote compute context. So within AML Workbench I'm taking advantage of the power of the cloud. I could also run this against Spark HDInsight clusters to do parallel processing.

AML allows all of this power and flexibility, and even tracks experimentation runs for me through Git, so we can now look at different models that I've run over time, and see which one is performing the best.

Now, once I've figured out which model is ideal, I actually want to deploy that to production. I can do that using the Azure command line interface. This is going to package up the model into a Docker container, and create the web service for me.

Now, as Scott mentioned, I can actually run this Docker container in the cloud at scale, or I can put it on-premises, or I can even deploy it to an IoT edge gateway. So this AI model could be run anywhere that supports Docker.

So with that, I'm actually going to turn back to my dashboard.

Now, when I'm actually clicking through here, and I click on an image, what I'm actually doing in this dashboard is sending the raw pixel images up to that web service, and scoring in real time to get the prediction back.

So in just a few minutes I was able to create a custom vision service.

Now, the best way to predict the future is to create it, and with Azure Machine Learning Jabil's revolutionizing their industry and creating the digital, intelligent, and predictive factory.

Thank you so much. Back to you, Scott.

(Applause.)

SCOTT GUTHRIE: We're really excited that everything Danielle just showed is now available for you to start using, both all the updates in the Azure Machine Learning service, as well as the new Azure Machine Learning Workbench that you can download and start using on any computer today to start building rich AI applications.

Today, we're also excited to announce a new VM family inside Azure that uses the latest P40 and P100 NVIDIA GPUs. This family of GPUs is optimized for deep learning AI scenarios, enabling you to build even richer AI applications, and Azure is the only public cloud provider today that supports both the P40 and P100, giving you even more flexibility as you look to build intelligent apps.

So the Microsoft Cloud is optimized for organizations. For us enterprises are not an afterthought, they're a critical design point. And it's not just about technology. You know, Microsoft has decades of experience supporting businesses and enterprise customers of every size. This means we really understand the critical requirements of running software for businesses, including certification, data sovereignty, security, and privacy. You know, in fact, Azure now has more compliance certifications than any other cloud provider out there.

With Azure we also give you the tools and integrated monitoring support to ensure that the workloads you put on Azure follow security best practices, and apply the comprehensive security intelligence that Microsoft has to your applications.

And we're releasing a bunch of great new capabilities to our Azure Security Center this week that's integrated directly into the Azure Management Console, including new hybrid threat detection capabilities, as well as security incident remediation support.

And what I'd like to do is invite Sarah onstage to demo some of these new capabilities, and show you how they enable you to build even more secure applications in the cloud. Here's Sarah.

(Applause.)

SARAH FENDER: Thanks, Scott.

With built-in intelligence and analytics, Azure Security Center can help every Azure customer be more secure.

Today, I'm going to show you how Azure Security Center can help you ensure that security best practices are applied, help you to lock down remote access, and enable you to act quickly should an attack occur. Let's get started.

So here in Azure Security Center, which you can access from the left pane in the Azure portal, you get a unified view of security across all your Azure workloads, and it's always up to date.

Let's take a look at compute. Here we see a list of recommendations for best practices for securing your Azure VMs. This includes things like making sure that your systems are up to date, that encryption is enabled, and that your Windows and Linux configurations are hardened.

We can click in to get additional details. Here we see a list of potentially vulnerable configurations. And you can click in to take action to remediate these.

Now, one important best practice is to limit remote access to your virtual machines. But this can be a bit problematic, because many of you require the ability to remotely log in for management purposes.

Security Center's new just-in-time VM access capability helps. It discovers VMs that aren't protected, and you can simply specify rules for how you want your users to connect, and Security Center takes care of the rest.

When a user needs access to that virtual machine, they can request it here in Security Center, or you using PowerShell. They'll simply specify how they want to connect to that virtual machine and for how long. And as long as that request complies with the rules that were set earlier, access will be automatically granted to that VM, so a user can get in and do their work.

So unfortunately, attackers will continue to innovate, and you need tools that enable you to respond quickly when that happens. Azure Security Center alerts you to threats that it detects using advanced analytics like machine learning, and tapping into Microsoft's vast global threat intelligence. Related alerts are combined into security incidents like the one that we're seeing here.

Now, every second counts when you're under attack, and we're very excited to unveil a new investigation experience that greatly reduces the amount of time and the level of expertise required to investigate.

Here we see a security alert. In this case a user didn't follow best practices, and that enabled an attacker to gain access to a VM via a brute forced attack against one of their management ports. So we can click on that compromised server to learn more.

Built-in queries make it easy to access related security events, and you can also conduct your own ad hoc queries over security data as well.

Now, here we see that a user, Abby Becker, has added herself as an administrator to this particular machine. That's a big suspect.

So if we go back to the graph, we can learn more about what this account has been up to. We see that there's a sign-in from an unfamiliar location. It looks like this account has been compromised. So you need to understand what else has this not-Abby been up to.

We can see that the account was used to sign into another server in the environment, and on that machine to execute a suspicious process, likely some form of malware.

New integration with playbooks makes it easy to automate response. You can take action to block Abby's user account or to quarantine this VM to prevent further damage.

Now, Azure is the only major cloud provider to offer all of these built-in security capabilities, and we're so excited today to announce the ability to now extend these capabilities beyond Azure to help protect workloads running on-premises and in other clouds. This greatly simplifies the process of managing security and protecting against threats across all of your hybrid cloud workloads.

You can simply open Security Center and the Azure portal to get started today. Thank you.

(Applause.)

SCOTT GUTHRIE: So another important aspect of trust in the cloud is understanding how much the cloud costs. You know, no one wants a surprise bill. And as you run more and more workloads in the cloud, it becomes even more important that you can accurately budget and forecast how much your cloud is going to cost.

Earlier this summer, Microsoft acquired Cloudyn, a leader in cloud-based cost management, and today I'm excited to announce that Cloudyn is now integrated into Azure, and that we're making available the cloud and functionality to all Azure customers for free.

(Applause.)

This means that you can now easily optimize your Azure spend, and get greater visibility and accountability of your Azure usage, and even implement policies like internal chargebacks and budgets in a really easy way.

Today, we're also announcing a new reserve VM instance model that enables you to guarantee VM capacity ahead of use, as well as achieve cost savings of up to 72 percent when running VM workloads inside Azure.

And as part of this we're also introducing with this the flexibility to exchange or adjust reserved instance purchases, enabling you to save even more money using the cloud.

And to show off some of these new cost management capabilities in Azure, as well as how you can model savings using reserved instances, I'd like to invite Julia White onstage. Here's Julia.

(Applause.)

JULIA WHITE: Thanks, Scott.

All right, now, a key value of the cloud is paying for only what you use. It offers great cost savings over on-premises approaches. However, we've definitely heard from all of you that you need more insight and visibility into your costs to make sure you can feel in control of all of your cloud spend. And that's exactly what we now have with Azure Cost Management, powered by Cloudfy. And what it does, it gives you real time visibility into every aspect of your cloud spend, and you can also create departmental level budget and allocations.

So let's take a look. So here I am in the dashboard, and you can see over there there's a new cost management blade option. Now, this will go live in the portal later this afternoon. And when I get there, I can just deep link with single sign-on right into my Azure cost management, powered by Cloudfy, dashboard.

So here in my kind of overview dashboard I see all the different views of spend, I can break down in a number of different ways.

Now, one of the areas of the largest spend is around VMs, so let me go ahead and drill down into my VM spend specifically. And you can see I'm getting a really nice view of all of my spend across what was going on, but then I have this big spike right here. Now, this might be a great thing because my marketing campaign is on fire and everyone is hitting my site, or it might be something I need to drill into and understand. I can just do that very simply right here in the cost management tools.

But I can also get cost management in a bunch of other really helpful ways and get that visibility, things like where are my application costs, what are my department level costs, and what am I spending on R&D versus production as an example.

So here you see I have all that broken down. I can just hover over, I can see what my production system looks like, I can see different applications within that. And then more importantly, I can take that and download it directly into an Excel, and I can pull it up. And with this I can create cost allocations for different departments in my organization and hold them accountable, and also give them the visibility to do that at a departmental level.

But if I go back to my cost management, one of the other capabilities is around budget management. So here I've put in what my budget is and then what my current expected run rate when I'm actually using is going to look like. And you can see here I'm on path to exceed my budget. So I need to work on some optimization.

Now, there are two key ways to do optimization with the cost management tools. The first one is to look at my CPU utilization. So here I'm getting a drilldown of exactly what my utilization looks like across my environment. And here I have a number of resources that are very highly utilized, and I might actually consider upgrading those to large instances. But I equally have things that are just zero to 25 percent utilized, and these are opportunities for me to downgrade to smaller sizes that are less expensive, and get that budget back into line.

Now, the other way to do optimization is what Scott just announced, which is our reserved instances that are coming to Azure. Now, once they're in Azure, you get this view here in the cost management tools that show me what it would look like if I did a standard pay as you go approach or if I purchase with reserved instances. And I can see where that break-even is based on the project that I have. And in this specific example it shows me that using reserved instances would save me about 33 percent over time, so I quickly get that cost benefit tradeoff on how to best manage my spend.

So overall we're really excited about how Azure cost management gives you full visibility and control over all of your cloud spend. As Scott said, no one wants a surprise bill, and this gives you comprehensive budget management across that.

So thank you so much. Back to Scott.

(Applause.)

SCOTT GUTHRIE: All the great capability that Julia just mentioned is now available for free for every Azure customer, and you're going to be able to start taking advantage of it later this afternoon once it goes live.

(Applause.)

So the opportunity to build applications that can change the world has never been greater, and each of you now has access to cloud resources that were unimaginable just a few years ago. And there really has never been a better time to be in technology.

And all of us on the Azure team are really looking forward to helping enabling your success, and we can't wait to see the amazing solutions that we know you're going to build.

Thanks so much, and have a great rest of Ignite.

(Applause.)

END