Cloud Computing: The Big Picture

by David Chappell

Course Overview

Course Overview

Hi everybody. I'm David Chappell. Welcome to my course, Cloud Computing: The Big Picture. I'm the principal of Chappell & Associates in San Francisco, California, and I'm convinced that the rise of cloud computing is among the most important trends of our time. I don't care what your role is, if you work in technology, cloud computing is affecting you. You might write code that runs in a cloud platform, like Amazon Web Services or Azure, or maybe your organization is building a private cloud that you'll need to manage. And whoever you are, I know that you use cloud applications. This course provides a broad introduction to cloud computing. I don't assume that you have any prior knowledge of this topic. The major things I'll cover are cloud applications, which are commonly known as Software as a Service, or SaaS, cloud platforms, and private clouds. By the end of this course, you'll know enough to go deeper, if you choose to, into more specialized cloud topics. You'll also be able to do a better job working with and making decisions about cloud computing. I hope you will join me to learn about this important topic with the Pluralsight course, Cloud Computing: The Big Picture.

Introduction

Introduction

The rise of the cloud is unquestionably one of the most important changes in the history of computing. I'm David Chappell, and in this course, I'm going to walk you through the big picture of cloud computing. To start, let's look at the world without cloud computing. An organization that's not using cloud computing looks something like this. It's got users and an on-premises data center, giving the organization its own compute, storage, and networking facilities. In this data center, they're running applications. Sometimes the applications run right on the hardware, but more often today, they're running in virtual machines, in VMs. This is the classic world of enterprise computing. This is how people have done computing since the dawn of the computing era. But more and more, organizations don't look like this. The rise of cloud computing brings big changes. It's useful to think about cloud computing in three big categories, three big ways that the cloud can change the world I just showed you. One of them is cloud applications, commonly called Software as a Service, or SaaS, which means running applications at data centers owned by somebody else accessible across the internet. The second category is cloud platforms, which include technologies for running applications, storing data, and more, again, running in data centers owned by somebody else accessible across the internet. And the third category, in most people's view, is private clouds. The general idea of private clouds is to take cloud platform technologies and run them in on-premises data centers. As we'll discuss later though, there are in fact various interpretations of what private cloud really means. And one more important idea, a term you hear a lot today is hybrid cloud. The core notion is that you can combine cloud platforms with private cloud technology to create a hybrid of both. I will say more about this later when we talk about private clouds. Here's how these three things change the traditional on-premises world. In the cloud era, we still see at least some organizations running applications on-premises, but we also more and more see organizations using SaaS applications accessed across the internet and running at some service provider. These cloud applications are running on compute, storage, and networking owned by somebody else. We also see organizations using cloud platforms, which provide services to build applications, work with data, and more. And finally, in at least some organizations, we're seeing the use of private clouds, which in the name are technologies from public cloud platforms that are brought in house. In this course, I'm going to talk about all three of these aspects of cloud computing, cloud applications, SaaS, cloud platforms, and private cloud. They're all important, so you need to know something about all of them. Ready to go? Let's get started.

Cloud Applications: Software as a Service

Introducing SaaS

Most organizations rely on some packaged software, applications they buy from somebody else. Increasingly though, that packaged software is running in the cloud. Rather than running software on-premises in their own data centers, organizations are using Software as a Service. In the diagram that I'm using to organize this course, cloud applications fit right here, but the picture raises an obvious question, doesn't it? The question is, well, what's the big deal? The idea of having applications that run remotely at some service provider that you access from your organization is hardly new. It's been around forever. People have run email servers, and SharePoint sites, and all sorts of business applications, and hosters, and elsewhere, so what's the big deal? How is SaaS different? Well, there are a few answers, but the most important is that traditionally, when a hoster provided an application, they ran an instance of the complete application just for your organization. In SaaS, that's not typical. SaaS applications are generally multi-tenant. Now that's a fancy word, but in fact, all multi-tenant really means is shared. It means that a single application is being shared by multiple customizers, multiple organizations. Each of these customers has its own data, and it's the responsibility of the application to keep that data separate and secure. Not every cloud application uses this approach, but it's fair to say that the idea of multi-tenancy is central to the modern notion of SaaS.

SaaS Examples

Lots of different vendors provide SaaS offerings across various areas. I'm going to discuss a few of them, looking at CRM, customer relationship management, email, storage, productivity, and ERP. Most folks would agree, I think, that Salesforce CRM is really the beginning of modern SaaS. It was Salesforce who showed us that SaaS worked as both a technology and a business. Their CRM offering remains a flagship SaaS technology today, but they're hardly alone. Traditional package software vendors are also offering SaaS for CRM and many other things. Oracle, for instance, offers CX Cloud for CRM. And for ERP, they purchased NetSuite, a pure SaaS startup, a while ago. Other traditional package software vendors are also here, such as SAP, who also offers CRM and ERP. Firms like Microsoft are also in this market in a big way. Microsoft's Dynamics 365 offers CRM, and ERP, and more. They also have SaaS solutions for email with Exchange Online, for

Storage with OneDrive, productivity with Office Online, and others that I'm now showing here. Google is a big SaaS provider too, and not so much in CRM or ERP, but they provide the very popular Gmail along with Google Drive for storage, Google Docs for productivity, and other solutions. Dropbox and Box are both offering storage today, and there are lots more. This is just a small sample of what is, in fact, a very large set of vendors offering SaaS solutions across a broad range of markets.

Evaluating SaaS: Users

Like pretty much everything, SaaS has pros and cons. From the point of view of a user of SaaS, there are benefits and there are risks to adopting this approach. The benefits include these. You can deploy the application faster because there is no local installation required. Unlike an on-premises application where you've got to put on your own servers, configure it, get it running, with SaaS, at least in a simple case, you can just point your browser at the app in the cloud and start using it. Now the reality is there might be some deployment work required with customization and so on, but still, SaaS applications are typically faster to deploy than on-premises software. SaaS also typically offers usage-based pricing, letting you pay only for what you use. It's typical, for example, to pay per user per month, not obligatory, but common. This lets you not overpay for software you don't actually use. It also makes it easier to scale up and down the number of users a given SaaS application has in your organization. SaaS also can bring less financial risk because the up-front cost is smaller. You can start small, only a few users, then grow if in fact it's useful. Products commonly have free trials, which means you can try the application at not cost at all. If you don't like it, if it has no value, just stop using it. Compare that to the conventional scenario of deploying a packaged application in your own data center. There, you buy the app, you install it, you deploy it, then you find out if it has business value. That's a big risk. It's much less risky with SaaS. SaaS, of course, also brings a reduced need for on-premises resources, like servers and IT staff, since SaaS applications are running in the cloud. Now obviously, that's a benefit unless you're one of those IT staff people who would have otherwise run on-premises applications. But from the organization's point of view, that's probably a benefit. SaaS also provides easier upgrades. Why? Because the SaaS provider does the upgrades, not you. Rather than installing updates periodically to keep the application up to date, the SaaS provider does that for you, providing you with new features every so often. SaaS has risks, as well as benefits though, like everything. It's important to grasp those before considering moving in this area. For example, SaaS requires that you trust your provider, your cloud provider, for availability and for security. This can be a big issue. Trusting your SaaS provider is very important. It can be

hard sometimes to build that trust. The reality though is that we have to build that trust to use these kinds of cloud solutions. And the truth is this, the truth is that if you think your own data center is more secure than the ones used by the major SaaS providers, you are almost certainly wrong. So chances are, using cloud solutions from large reputable organizations will improve your security, not reduce it. Still, SaaS can raise legal and regulatory concerns because you're storing data outside your own environment. In fact, sometimes, SaaS applications will be running in some other country, which means you're storing data outside your national borders. That can be an issue. We're far enough into the cloud revolution that that's been dealt with in many, many, many scenarios, but it's still something to be aware of. SaaS can also limit customization. I told you a few minutes ago, one of the hallmarks of SaaS, typically, is multi-tenancy, which means lots of customers are sharing the same app. This limits how much customization you can do compared to having your own instance of the code. Still, modern SaaS providers have found a way to get around this through customization on the side of the app, through configuration, or in other ways. Integration can also be harder with SaaS applications. Integration is hard with on-premises applications, so with SaaS apps in the cloud, there is even more work to do. Again though, we're far enough into this cloud world that integration with SaaS has largely become a solved problem. And finally, sometimes SaaS applications can have lower performance than on-prem applications. Why? Well, the big reason is the network hop. There are some scenarios where that's just too slow, too much to take. For example, a real-time factory floor environment might find the, I don't know, third of a second delay going to a SaaS application across the internet too much to bear. So there are situations where the delay created by the fact that SaaS apps run across the internet can be too much. Like everything, SaaS has benefits, SaaS has risks for users. Most people today most often seem to find that the benefits outweigh the risks.

Evaluating SaaS: Software Vendors

Just as SaaS has both benefits and risks for users of software, SaaS has benefits and risks for software vendors. Among the benefits are these. SaaS can provide a more predictable revenue stream than traditional licensing. Rather than the big chunks of money you receive each time you sell a license, a SaaS vendor gets per users, per month revenues. That predictability is attractive for a bunch of reasons. You can plan better, and investors, Wall Street, likes it better as well. SaaS also lets vendors sell directly to business decision makers without going through IT. Since it's the business people who are usually buying the app, they're the ones who want it, to be able to sell just to them and not have IT be a blocker can be a huge win for software companies. Since IT is no longer running the application, they indeed not have much of a say, potentially, in which one is

chosen. Sales people for SaaS vendors typically like this. Going to SaaS can also lower support costs because you've got this shared multi-tenant application that you're maintaining. You don't have to maintain many, many copies of your software on customer premises. That's a good thing. SaaS also can provide more knowledge about how customers use a vendor's application. The vendor can actually see what's going on, can see directly which customers are doing what. That's really useful for things such as knowing how each customer uses your application. And finally, SaaS can offer the potential to reach new customers in broader markets Because, typically, it's just a website. People can find you, and buy you, or at least try you, directly. You can reach new customers with less effort. The risks though of software vendors adopting SaaS are substantial. They include things like this. The vendor must now demonstrate real value up front, and the reason is the try before you buy option that's typical in SaaS applications. In the conventional on-premises packaged software world, the customer has to take on faith that there will be business value, so slick sales people can be very convincing. With SaaS, the customer actually tries the application. They know whether or not it has real business value. Also, revenue comes in more slowly because of the typical SaaS pricing models. Per user, per month can be a great thing over time for SaaS vendors' bottom line, but it takes longer to get revenue coming in than with selling big licenses up front. SaaS can also lessen a vendor's ability to sell customization services, because as I mentioned earlier, multi-tenant applications can be harder to customize. This isn't always true, but some vendors that make a lot of money on services have found that moving to SaaS can hurt that part of their business. SaaS can also bring new sales challenges, such as customers who are just not cloud friendly. In fact, especially for smaller software companies, it's almost obligatory in some cases to have both a SaaS and an on-premises version of your application because some customers will insist on each. Finally, the biggest and most general risk is that going to SaaS requires really substantial business changes for on-premises software companies, how they price, how they sell, how they pay commissions, all these things commonly have to change. It is a big decision. It's a big change for a software company. Still, for most package software companies today, do they have a choice? Can they feasibly remain selling only on-premises software? Probably not. The reality is SaaS more and more is ruling the world of purchased software.

The Impact of SaaS

Let's summarize. What's the impact of SaaS? Well, it depends on whether you're a software user or a software vendor. For users, life is better, at least as long as the risks aren't show stoppers. Life is better because you need no longer install, and operate, and maintain, and patch on-

premises software applications. You need no longer take the giant risk of writing a big check up front for an application that you hope will have business value. With SaaS, the vendor runs the software and takes the burden of operations off you, and the vendor, typically, lets you try before you buy so you know whether or not the app will have value. There are huge benefits for users. Now sometimes, the risks are too great. Sometimes, for example, regulatory concerns will stop you from using SaaS, but that's not true, however. The great majority of users seem to find that they prefer SaaS to conventional on-premises software. For software vendors, life is different. Is it better? Is it worse? It's unclear, but it's certainly different because everything changes, pricing, the sales process, operations. I didn't mention this earlier. It's important though. In conventional on-premises software, the user is burdened with running everything. They own the computers. They do operations. They do management. In the SaaS world, That burden shifts to the vendor, and so software vendors now have to learn to effectively run large scalable applications. This is a whole new skillset, but being successful at SaaS requires that you get good at this. So for software vendors, the impact of SaaS is both positive and negative, pros and cons, but it is certainly huge. Everything changes. The main points are these. SaaS brings big changes to software users, to software vendors. SaaS has pros and cons, but the pros outweigh the cons in a majority of situations. This is why SaaS is remaking the software industry.

Cloud Platforms

Introducing Cloud Platforms

Cloud platforms are an essential aspect of cloud computing. In fact, to developers, people who write software, they're the most important part. Before I talk about them, let's review again the big picture of cloud computing. Even in the cloud era, many organizations will continue to run applications on-premises, but more and more, they'll rely on external service providers that run cloud applications, SaaS applications, and provide cloud platforms. Organizations will also, in some cases, bring cloud platform technologies on-premises in the form of private clouds. Our focus in this module is on cloud platforms. There's an obvious question here though. Cloud platforms provide services like virtual machines, and storage, and so on through a remote provider accessed across the internet. But wait a minute, isn't that called hosting? Understanding what cloud platforms offer over traditional hosting is important. First, cloud platforms offer much more immediate access to a broader range of services than conventional hosting providers.

Unlike a classic hoster where you might get a machine of your own, one that might take a while to install, cloud platforms let you get a virtual machine, or a database, or something else through your browser in just a few minutes. It's quite different. Also, cloud platforms today offer a very wide range of services, as we'll see, far more than conventional hosting. Another big change brought by cloud platforms was the advent of usage-based pricing, letting you pay for what you use and no more. Rather than having a server dedicated to you full time, you could instead rent virtual machines by the hour or by the minute, rent data storage by the gigabyte, and so on. Usage-based pricing was a real innovation of cloud platforms. And finally, the global scale that the major cloud platform providers bring is phenomenal. Their data centers are much bigger than conventional hosters. This scale provides a number of benefits, such as the ability to keep costs very low. So while in some ways, cloud platforms are similar to traditional hosting, in some ways, they're an evolution of hosting, they're still different in many important aspects.

Evaluating Cloud Platforms

Evaluating cloud platforms requires looking at both the benefits and the risks. Let's start with benefits. First, cloud platforms give you faster deployment because there's no wait for computing resources. You don't need to order a physical server and wait weeks for it to be delivered and installed. You could instead through your browser ask for virtual machines, or data, or something else and have them available in minutes. Also, cloud platforms give you usage-based pricing as I just discussed. You pay for what you use. This, in turn, brings less financial risk because you now have much lower up-front investment in hardware and software. You don't have to buy servers and hope that whatever you're doing, your project, your startup, whatever, is successful. You need not make that large up-front investment before you can do anything. Instead, you rent what you need. If the project succeeds, great! If it doesn't, shut things down and stop paying. This lower financial risk for new projects means that you could do more innovation. You can take more chances because you're spending less on each attempt. That's a big win, and it's a big reason why cloud platforms have helped create the boom in IT startups that we've seen in the last few years. Cloud platforms also bring a reduced need for on-premises resources, such as servers and IT staff. Now that's a benefit, unless you're one of those IT staff, separate issue. But for developers, for organizations as a whole, having fewer on-premises resources to pay for is a good thing. Cloud platforms also give you easier upgrades because you no longer have on-premises software to update. There are some real benefits to cloud platforms, but there are also some real risks. For most people, the first thing on the list of risks is the same thing that was first on the list of risks for SaaS applications. It's the challenge of trusting a cloud provider to be there, to be available,

and most of all, to keep your data secure. Let's talk about this for just a minute because it's really important. I mentioned in the last module that if you think a cloud provider's data platform is less secure than your own, you're probably mistaken, at least for the major providers. But more than that, we are not entirely rational in how we trust. Ask yourself this, what is the world's most dangerous technology? What is the world's scariest technology? Is it nuclear weapons? It is some computer virus that's going to come wipe all our machines? No. No, the world's scariest technology, the world's most dangerous technology is Windows Update. Because every patch Tuesday, Microsoft, through Windows Update, puts whatever software it chooses directly into our operating systems, and we all just say, sure, install. How do we know that the next update or the last one won't steal all of our data? The answer is that we don't, but we trust Microsoft. The benefits of these updates are substantial, and we've learned to trust Microsoft. I'm not suggesting that Windows Update is actually dangerous. Rather, I am suggesting that over time, we learn to trust organizations that provide useful services and prove themselves worth of that trust. That's exactly what's happening with cloud platforms today. More and more, we trust them as we've long trusted our vendors for many other things. Another risk, also similar to what we saw with SaaS applications, is that cloud platforms can raise legal and regulatory concerns. Increasingly, however, as cloud computing becomes the norm, this is being dealt with through regulatory change or other kinds of understandings. Integration can also be harder. Integrating applications is challenging when they're all in the same data center. When they're in the cloud, it can be even tougher. Although, once again, the maturing of cloud computing makes this simpler over time. Sometimes, lower performance is a concern. There are situations where cloud platforms just aren't appropriate for performance reasons. For example, an application controlling real-time controls on a factory floor might not be able to accept the delay, the latency, of talking to a public cloud application. In fact, that's one reason why organizations use private clouds in some situations. And for developers, cloud platforms can give them less control. On-premises, developers can have physical machines they own entirely to work with. In the public cloud, however, you've got virtual machines, or sometimes, as we'll see, even a higher level of abstractions to work with. Developers have to learn to live within the constraints that those abstractions provide. So like almost everything in life, cloud platforms have benefits, and they have risks. In many situations, increasingly, in a majority of situations, the benefits outweigh the risks.

Cloud Platform Technologies

So far, I've talked about cloud platforms in general terms. To understand this area though, we've got to also look at cloud platform technologies. I'm going to start with compute technologies. The most fundamental cloud platform compute technology is called Infrastructure as a Service, IaaS, pronounced IaaS or EaaS depending on where you live in the world, and the idea is simple. As the figure shows, a user, such as a developer, can create and use virtual machines. The cloud platform provides VM images that can be used to create those VMs, and the leading platforms all support both Windows and Linux. This is useful for all kinds of things. A developer, for example, might build a new application in IaaS VMs, or you might move an existing on-premises application, either a packaged app or a custom app, from your on-premises data center onto a cloud platform. They're just virtual machines. So just as apps can run on-premises in VMs, they can often, although not quite always, run unchanged in IaaS VMs. IaaS is really useful, but it's not the only choice. It's also possible to build and run applications using an approach called PaaS, Platform as a Service. Here, applications are still running in VMs, but the developer who writes those applications doesn't see the VMs. Instead, he or she is running the application on some set of supporting services. What those are can vary quite a bit. But across all PaaS technologies for compute, the services make it easier and faster to build and run applications on cloud platforms. It also can make it cheaper to run these applications because PaaS lowers the amount of management work required. The PaaS services take care of a lot of the housekeeping for you. PaaS is designed mostly for brand-new applications, and for that, it can be a great choice. For quite a while, IaaS and PaaS were pretty much the whole story for cloud platform compute. Today though, there are more options. For example, there is what's sometimes known as Containers as a Service. The idea is that rather than working at the virtual machine level, you instead can deploy and manage containers that run inside VMs then have your applications run inside those containers. A container management service can decide where your containers should run and can manage them as they run. The most popular choice for this service today is called Kubernetes. Kubernetes was originally built by Google, but it's supported today by all of the leading cloud platforms. Another options for compute on cloud platforms is an idea that's sometimes called Functions as a Service. It's also commonly referred to as serverless computing. The idea is that you need not think about virtual machines or even containers. Instead, a developer can just build applications as functions and invoke those functions directly. This is simple, which is good, and it can also be very cheap because you're typically charged based only on execution time, which is often quite small. You can wind up having a powerful application that handles lots and lots of requests from many users for not much money. Another category of cloud platform technologies are those for working with data. Among the most important approaches here are these. You're always free to run a database management system such as

SQL Server, or MySQL, or Oracle in a virtual machine, an IaaS VM; however, cloud platforms also provide managed services for working with data, for example, most of them provide object storage. And despite the name, object storage just provides unstructured data, raw bytes. You can also use cloud services for relational storage or for NoSQL storage, such as document-oriented JSON databases. All of these are managed services, which aren't the same as running your own database in a VM. Managed systems can be simpler, less expensive, and easier to operate than running a standard database system in a VM. They can also scale much better to handle more users and more data. To think about all of these cloud platform offerings together, it's useful to create a grid that shows the leading technologies and the major vendors in this market. In this grid, the columns are compute technologies, IaaS, PaaS, Containers as a Service, and Functions as a Services, followed by data technologies, object storage, relational, and NoSQL. The rows will be cloud platform vendors, Amazon Web Services, Microsoft Azure, and Google Cloud Platform. Are there other firms in the cloud platform market? Sure, absolutely. Are these three the leaders today? They are, and so they're the ones most worth looking at in this course. Next, I want to take a look at each of these three vendors using this grid to summarize what they offer.

Example: Amazon Web Services

I'm going to start our walk through today's leading platforms with Amazon Web Services. AWS was the first real cloud platform, and it offers technologies in all the areas I've just talked about. For IaaS, for example, Amazon Web Services offers Elastic Compute Cloud, EC2, for PaaS, their primary service is Elastic Beanstalk, and for Containers as a Service, they offer the Elastic Kubernetes Service, EKS. As I mentioned earlier, all of the leading cloud platforms today support Kubernetes for working with containers. They also offer serverless computing, Functions as a Service, with a service called Lambda. For data, Amazon's object storage technology is S3, the Simple Storage Service. Their primary managed relational service is RDS, the Relational Database Service, and their primary NoSQL Technology as a Service is DynamoDB, although there are others as well. Some things to know about Amazon Web Services, first, they were the first mover. AWS first appeared with S3 and then EC2 in 2006, quite a while ago. They're viewed as a leader for good reason. They have the largest market share today in cloud platforms. Also, AWS offers diverse services, a mix of open source things, like Kubernetes and MySQL, with proprietary solutions, including Microsoft Windows and lots of things invented by Amazon itself, such as DynamoDB.

Example: Microsoft Azure

Another major player in the world of public cloud platforms is Microsoft Azure. Azure also offers technologies in all the categories I've introduced so far. Their IaaS technology is called Virtual Machines, and their primary PaaS offering is App Service. For containers, Microsoft provides Azure Kubernetes Service, AKS, and for serverless computing, they've got Azure Functions. For object storage, Azure has Blobs, for a relational service, they've got SQL Database, and for NoSQL, there are various options with Cosmos DB as perhaps the most visible. Some things to know about Microsoft Azure include this, Microsoft is a technology company. Unlike Amazon, which is in many different business, retail, and movies, computing, and lots more. Microsoft has chosen not to become a conglomerate. They have remained focused on technology. Microsoft also has strong relationships with many existing enterprises because Microsoft is a core enterprise vendor and has been for a long time. In part because of this, Microsoft has focused for quite a while on Hybrid Cloud. They very much encourage customers to combine their on-premises resources with the public cloud. Also, like AWS, Microsoft offers diverse services, providing both open-source and proprietary technologies on their cloud platform. In fact, something like half of the VMs running on Azure today run Linux, not Windows. Microsoft is clearly a different company today than it was just a few years ago.

Example: Google Cloud Platform

The third major vendor in this market is Google, with Google Cloud Platform. For IaaS, Google offers Compute Engine, for PaaS, App Engine, for container support, Kubernetes Engine, and for serverless computing functioning as a service, they offer Cloud Functions. For data, Google offers Cloud Storage for object storage, Cloud SQL as a managed relational service, and Cloud Bigtable for a managed NoSQL technology. It's worth noting here that all three of the major cloud platform vendors have similar technology sets. This is no surprise. It's what commonly happens as markets mature. Just a few years ago, these three offerings were much more different from each other, but today, they've all worked out what the market really wants. And so we've seen this convergence of services. That's why it's possible to think about the core Cloud Platform services in a simple grid like this. Some things to know about Google Cloud Platform include these. Google was later to the market. Amazon, then Microsoft, were first with a fully featured cloud platform. Google today seems wholly committed, however, to catching up in this space. As part of that, they can be a price leader. They have said that cloud pricing should follow Moore's law, which is a very aggressive statement, and it cut prices quite often. Now, the other cloud platforms follow them, so it's hard to have a sustained price advantage. But still, it's a position that Google has

expressed. Also, like Amazon and Microsoft, Google Cloud Platform offers diverse services, including open source and proprietary options. They too support both Windows and Linux together with services that are unique to the Google platform. Please, don't be confused, however. Even though we've just talked about a core set of technologies that are common across all three cloud platform leaders today, there are many other services offered by these platforms. For example, support for big data processing with Spark and Hadoop is present in all three of them. Amazon has EMR, which originally stood for Elastic MapReduce, Azure has HDInsight, and Google has Cloud Dataproc. All three platforms also offer machine learning technologies, Amazon ML, Azure ML, and Google AI Platform. And there are a lots more. I'm showing you the fundamental services, but the three leading platforms all offer dozens more.

Other Cloud Platforms

The big three cloud platforms, Amazon Web Services, Microsoft Azure, and Google Cloud Platform, get most of the attention in this area, but there are others, and they're worth mentioning. For example, Alibaba Cloud offers IaaS, object storage, Kubernetes, and lots more. They offer a full suite of cloud services, and they are a dominant player in China in particular. IBM Cloud also offers IaaS, object storage, Kubernetes, lots of cloud services. Interestingly, they also offer bare-metal servers. You can actually rent just raw hardware from them. And there are lots more. For example, many traditional hosters now offer some cloud services. The challenge, though, is that they don't have the scale of the market leaders, and so it's tough for them to compete in price and other areas. The key idea to keep in mind is that while the big three have the majority of market share today, there are plenty of other smaller vendors in this market as well.

Low-code Platforms

The three cloud platforms I've just talked about, AWS, Microsoft Azure, and Google Cloud Platform, all provide a broad range of services, but there's another kind of cloud platform that also fits in this discussion, one that focuses on a very specific thing, helping people who might not be professional developers build new applications quickly and easily. These offerings are commonly known as low code development platforms. It's fair to think of these as a flavor of PaaS, a Platform as a Service. A low code platform provides services for creating application logic, working with data, and other things that people building applications need. But the PaaS services I talked about earlier in AWS, and Azure, and Google Cloud Platform are all aimed at

professional software developers. That's generally not true of low code platforms. Pro developers certainly can use them, and they often do, but the real target for most low code platforms is what are known as citizen developers, people who have some technical knowledge, but aren't professional software developers. The goal is to help more people build more applications so organizations can get more apps up and running with less effort. And low code platforms are widely used today. The popular choices include Lightning from Salesforce and Microsoft's PowerApps. It's certainly fair to say that the low code approach is an important aspect of cloud platforms today.

The Impact of Cloud Platforms

There are lots of ways to think about the impact of cloud platforms. For startup companies, they've been transformative. It's now much cheaper and much easier to do a technology-based startup because you can use public cloud platforms rather than buy your own hardware. This is a big win. And even for enterprise IT, cloud platforms are having a huge impact. Here's one way to think about that. Traditionally, the enterprise IT default platform looked like this. The clients were PCs and laptops. Everyone had their own internal data center infrastructure with servers, and operating systems, and databases, and so on. On top of that, organizations run packaged applications like SAP, and SharePoint, and whatever, and their own custom apps. This was the default enterprise world for decades. In the world of cloud though, this changes. First, as we all know, the clients have changed. PCs and laptops still matter, but so do mobile devices, like tablets and phones. And the server-side is changing radically for most organizations. It now looks like this. Whenever possible, organizations are choosing to use SaaS rather than on-premises applications. More and more, on-premises packaged apps are being replaced by apps in the cloud. Similarly, custom applications that were once written on an organization's own internal infrastructure are now built on cloud platforms. Cloud platforms are the new foundation for competitive advantage. Why is that? It's because an app you can purchase, whether it's an on-premises packaged app or a SaaS app, can be bought by your competitors as well. No real advantage is usually possible here. Real advantage comes from custom applications that you build yourself that do unique things. That's what gives you competitive advantage, and that is now the province of cloud platforms. That's where you'll now build your custom apps. The choice your organization makes here is very important. The main points of this module are these. Cloud platforms have pros and cons like everything. In a majority of situations though, you should expect the pros to outweigh the cons. And the leaders in this market today are clear, Amazon

and Microsoft with Google maybe a little behind, and in low code platforms, Salesforce. And finally, the big truth, cloud platforms are becoming the default for new custom applications.

Private Clouds

Introducing Private Clouds

Some people think that private clouds are the least important part of cloud technology. Others think they're essential. Whatever the truth, they're certainly worth talking about. Remember what the big picture looks like. Many organizations used in cloud computing are likely to still run some applications on-premises, at least for a while. They will, however, use service providers to provide cloud applications, SaaS, and cloud platforms. Private clouds, our topic here, bring the technology of the cloud into an organization's own on-premises world. Exactly what that means is up for debate, however. The traditional It's taking public cloud services and providing them on-premises. However, as we'll see, there are various perspectives on what that actually means. One thing that's clear, however, is what private cloud is not. It's not just a new name for traditional on-premises data centers, but the term is sometimes used this way. A CIO might say, well, we're going to run on our private cloud when all they really mean is their own on-premises data center. This is not right. Private cloud to mean anything at all has to mean bringing some cloud technologies into your own on-premises world. It's just not fair to begin referring to your on-premises data center with no changes as a private cloud. It is fair to say, however, that private cloud is a confusing market. It's a confusing market largely because vendors use the term in different ways as we'll see. Most conversations today about private cloud are also about hybrid cloud, so what is a hybrid cloud? The first thing you might think of is, well, if I've got an application running in the public cloud with data on a private cloud, that must be a hybrid cloud. But that's not right. That's a hybrid application. Any public cloud platform can do hybrid applications, so don't confuse a hybrid application with a hybrid cloud. To be a hybrid cloud, you need to have public cloud services linked with private clouds in some way, and various services can be linked. So for example, you might have compute services like IaaS that work in both the public cloud and a private cloud and work together in some fashion. You might have compute services like PaaS that let you move applications back and forth. You might have common idenity, or common management, or other tools. The point is, hybrid cloud is an important ID, and it means combining public and private cloud services. It's not the same thing as a simple hybrid

application. I'm going to talk about three approaches to private cloud, all of which are different and all of which have value. I'll talk first about the idea of deploying public cloud technology on on-premises servers. Examples of this include Microsoft Azure Stack and Amazon's AWS Outposts. I'll also talk about OpenStack, which is open-source software offering cloud services for on-premises servers. And finally, I'm going to talk about using cloud technology for creating Internet of Things, IoT devices, that link with public cloud platforms. Examples of this include AWS IoT Greengrass and Azure IoT Hub. All three of these approaches are interesting, and all three are worth looking at.

Example: Microsoft Azure Stack

For the first approach to private clouds, putting public cloud technology on on-premises servers, I'm going to look at Microsoft's Azure Stack. It's simple to understand. Here's the idea. As we discussed earlier, Azure provides services in the public cloud. Azure Stack takes a subset of those services and makes them available in enterprise data centers and at hosters. Azure Stack bring public cloud services to on-premises data centers and hoster data centers. What kinds of services? Things like IaaS with Azure Virtual Machines, VMs on demand, PaaS services such as Azure's App Service, Containers as a Service with Azure Kubernetes service, Functions as a Service, serverless computing with Azure Functions, and more, like Blobs for object storage. Some things to know about Azure Stack. First, it's really Azure. It uses a subset of Microsoft's Azure code. Second, it's sold as an appliance. You don't buy just the software. You buy the hardware and the software packaged together from a vendor like HP Enterprise, or Dell, or somebody else. This might seem odd, but the truth is that making this cloud software work effectively on any old random heterogeneous hardware you might have lying around is difficult. So instead, Microsoft defines a hardware you have to use with Azure Stack. This makes it much more like Azure data centers, which themselves have quite homogeneous sets of hardware. And one more thing to know about Azure Stack is that it changes organizations because management of an Azure Stack environment differs from how you manage conventional servers. With conventional servers, you commonly have the server people, and storage people, and networking people, but Azure Stack provides all of these as a unified set of technology. And so organizations that adopt this solution will probably need to change how their people are organized too.

Example: Open Stack

Another approach to private clouds is to deploy open-source software. It implements cloud services on your on-premises servers. The most visible technology for doing this today is OpenStack. OpenStack provides cloud platform services for enterprises and hosters. This is kind of like Azure Stack. But a big difference from Azure Stack is that while Azure Stack is derived from the services of a major public cloud, Microsoft Azure, OpenStack doesn't really have much public cloud support today, and OpenStack is just software. It needn't be deployed as a combined hardware/software appliance as with Azure Stack. OpenStack provides services such as IaaS using a technology called Nova by OpenStack, and it provides object storage with Swift. It also provides lots of other things, including software called Sahara for managing Hadoop clusters and more. Some things to know about OpenStack include these. First, it was born in the public cloud. Rackspace, a large hoster, was one of its creators, and the original intent was to make OpenStack work effectively in public and private clouds. What's happened, however, is that there's limited public cloud usage of OpenStack today. None of the big three public cloud platform vendors support OpenStack. Amazon, Microsoft, and Google all have their own public cloud technologies; however, there is diverse vendor support for private clouds using OpenStack from Dell, from Red Hat, and others. It remains a piece of the cloud story. Still, it's fair to say that OpenStack hasn't been as widely adopted as its creators hoped it would be.

Example: AWS Greengrass

The third approach to private clouds that we're going to look at is using cloud technologies in on-premises IoT devices. The example I'm going to use for this is AWS IoT Greengrass. One could argue that Greengrass doesn't really fit in this section, that it's not really a private cloud, and that's actually a reasonable argument. Nonetheless, here's the idea. Amazon Web Services is a major public cloud platform. Greengrass takes services from AWS, the public cloud, and brings them into the enterprise. In that sense, it's a private cloud. The big difference though is that the technologies Greengrass includes are for IoT devices, not servers, and they can also be used in consumer contexts. What's interesting here is that Amazon is bringing services designed for the public cloud into an environment intended for a very different purpose. Those services include these. The key idea, the key technology in Greengrass is Amazon's Lambda. It provides Functions as a Service serverless computing. Greengrass also includes messaging services between Greengrass devices and more, but those are generally not borrowed from AWS and the public cloud. Rather, they're designed just for Greengrass. The big borrowing from the public cloud is Lambda, the serverless approach to computing. Some things to know about Greengrass include these. It's a different approach, and it addresses a different problem than Azure Stack and

OpenStack. In fact, it's very much aimed at the future. IoT is a high-growth market, while on-premises servers are not a terribly high-growth market today. In fact, so much new computing is in the public cloud that on-premises servers are declining. And why focus on private cloud for a shrinking number of systems? Why not focus instead on what will be left on-premises, such as IoT devices, once the public cloud becomes dominant? This argument makes some sense, and it explains why a technology like Greengrass exists. Greengrass is not a conventional private cloud technology, but it's nonetheless an interesting approach to using cloud-derived technologies in an on-premises environment.

The Impact of Private Clouds

What's the impact of private clouds today? How should you think about them? Well, one thing to realize is that you should think and, not or. Public and private clouds address different problems, and so they're complementary. Public clouds help you scale, help you get cheaper resources, and more. Private clouds make your on-premises world better. And remember, public cloud plus private cloud yields hybrid cloud, a common goal today. Also, the private cloud market has been slow to take off. This is due in part to complexity. A number of vendors tried to turn their on-premises virtualization environments into private clouds, something that didn't work out all that well, so we're now seeing other approaches. And finally, realize that private clouds can provide real value. For example, fast access to computing resources is a real win in many environments, and IaaS VMs on demand can give you that in your own data center. Just as important, private clouds give you modern application services. You can no longer just install the new version of Windows Server or Linux and get the latest application platform technologies. If you want things like serverless computing, you need to use a cloud platform. Those services are only available as part of cloud technologies, and so private clouds have become essential for getting modern application development services in your own environment. Especially if you're still building on-premises applications, don't ignore private clouds.

Summary

And there you have it, the big picture of cloud computing today. The key points to remember are these. First, cloud computing is here. It's a reality. It's a very important change in how our world works. Second, all three aspects of the cloud are worth understanding, cloud applications, SaaS, cloud platforms, and private clouds, including hybrid cloud. And finally, a new world is unfolding. This level of change happens so rarely in our industry, and yet, it's happening right now with the

growth of cloud computing. It's happening at a time when you and I are both working in this field. Aren't we lucky? It's a great time to work in technology. I'm David Chappell. Thanks for watching!

Course author

David Chappell

David Chappell is Principal of Chappell & Associates in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better...

Course info

Level	Beginner
Rating	★★★★½ (490)
My rating	★★★★★
Duration	1h 4m
Updated	13 Jan 2020

Share course