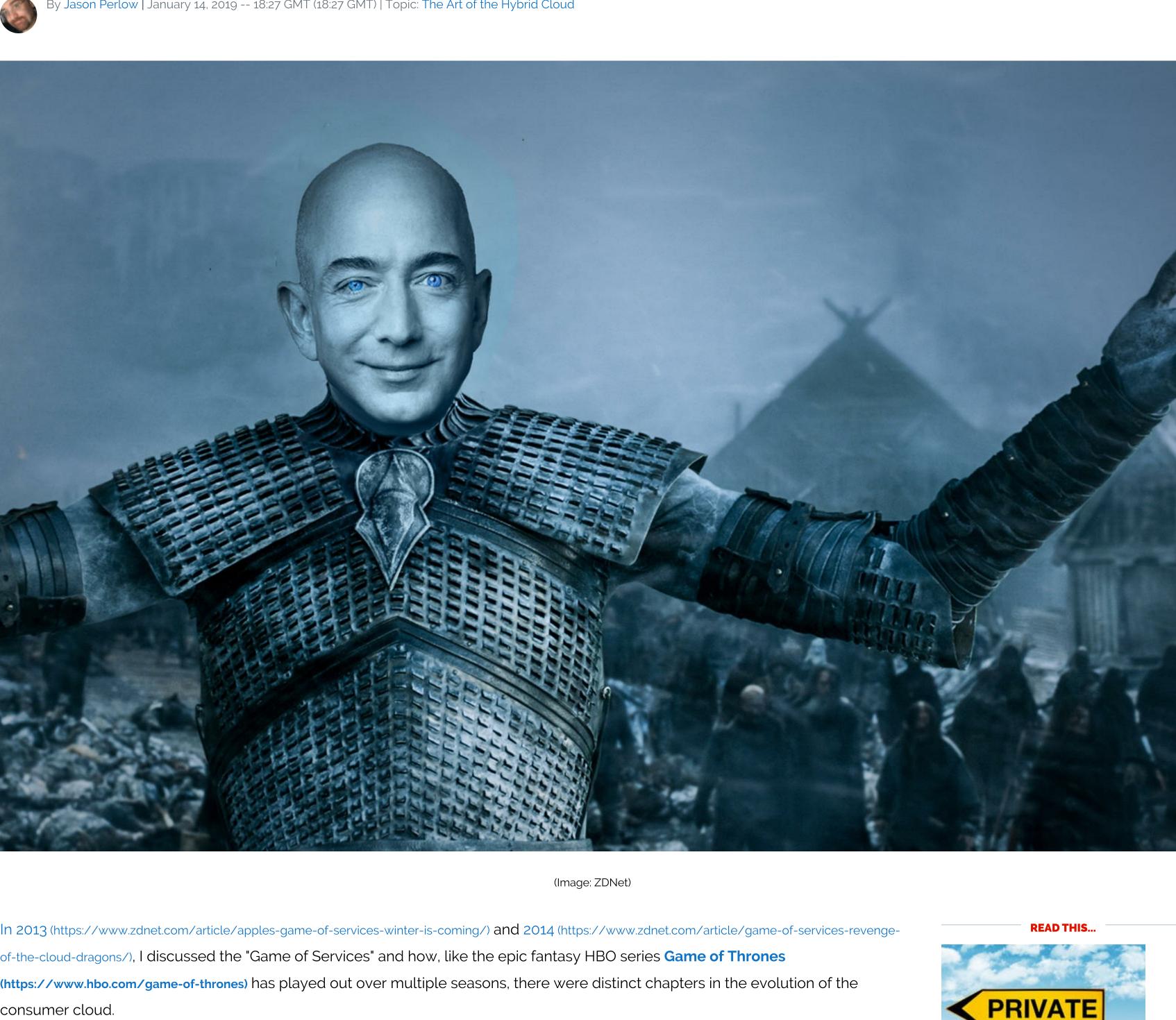# Game of Clouds: Lock-In is Coming

"Born in the Cloud" workloads might be designed to be platform-agnostic now, but that doesn't mean it will always be the case, as hyperscalers like Amazon Web Services and Microsoft Azure continue to add specific and unique functionality to their hosted services.

By Jason Perlow | January 14, 2019 — 16:17 GMT (16:17 GMT) | Topic: The Art of the Hybrid Cloud

Image: ZDNet

In 2013 https://www.zdnet.com/article/apples-game-of-services-winter-is-coming/ and 2014 https://www.zdnet.com/article/game-of-services-revenge-of-the-cloud-dragons/, I discussed the "Game of Services" and how, like the epic fantasy HBO series Game of Thrones (https://www.hbo.com/game-of-thrones) has played out over multiple seasons, there were distinct chapters in the evolution of the consumer cloud.

The evolution of the consumer cloud has been tied to the services that the major industry players https://www.zdnet.com/article/top-cloud-providers-2018-how-aws-microsoft-google-ibm-oracle-alibaba-stack-up/ — Apple, Google, Microsoft, and Amazon — are hosting on them and how they differentiate from each other.

The first chapter of Game of Services was about the mobile OS platforms, and which would be left standing. We now know who won: It's Apple and Google.

The second chapter was all about the services themselves, what the houses had to offer in terms of cloud storage, apps, messaging, and music/video/books content, and how those would continue to evolve.

The third chapter was about the battle over the API access to those public consumer cloud services. Google and Facebook, for the most part, are still the most important consumer clouds and access to those APIs are a constant moving target.

What's the fourth chapter? The evolution of the enterprise/commercial cloud. We are talking about the public cloud hyperscalers here: Amazon Web Services, Microsoft Azure, Google Cloud Platform, and to a lesser extent, IBM Cloud.

Up until now, enterprise cloud workloads could be classified as IaaS (Infrastructure as a Service), PaaS (Platform as a Service), or a mixture of all three. Workloads that are designed to run in the cloud from day one are often referred to as "Born in the Cloud" workloads, whereas traditional client-server workloads that originate from on-premises data centers, which are moved to the cloud, are migrated cloud workloads.

And, if you combine these things, splitting between cloud and on-premises, you have hybrid cloud architectures.

Also: Watch out for MongoDB, AWS launches fully-managed document database service https://www.zdnet.com/article/aws-launches-fully-managed-document-database-service/

MUST READ

- **Top cloud providers 2018: How AWS, Microsoft, Google, IBM, Oracle, Alibaba stack up** (https://www.zdnet.com/article/top-cloud-providers-2018-how-aws-microsoft-google-ibm-oracle-alibaba-stack-up/)
- **What a hybrid cloud is in the 'multi-cloud era,' and why you may already have one** (https://www.zdnet.com/article/what-a-hybrid-cloud-is-in-the-multi-cloud-era-and-why-you-may-already-have-one/)
- **We found 22 cloud services your business definitely needs to try** (https://www.zdnet.com/article/best-cloud-services-for-small-businesses/)
- **91% of tech leaders say hybrid cloud is 'ideal' IT model** (https://www.techrepublic.com/article/91-of-tech-leaders-say-hybrid-cloud-is-ideal-it-model/?ftag=CMG-01-10aaa0d) (TechRepublic)

For the most part, enterprise/commercial clouds have hosted services that could be deemed commodity or interchangeable. Compute-VMs, Storage, and networking are fundamental bread-and-butter IaaS services that are effectively the same among the major hyperscalers, and among the major players they have been priced the same, in race-to-the-bottom fashion.

Containers that run on open-source application packaging standards such as Docker https://www.docker.com/, and which use orchestration systems such as Kubernetes https://kubernetes.io/ are the next generation of compute — and that, too, is also becoming commoditized. It's cheaper and much easier to scale out than VMs. If you have a Docker-based application on one public cloud, it's fairly trivial to port it to another public cloud with similar containerization hosting services.

Also: What is the Kubernetes hybrid cloud and why it matters https://www.zdnet.com/article/what-is-the-kubernetes-hybrid-cloud-and-why-it-matters/

The hyperscalers can try to differentiate on performance and a few other things, but at a fundamental level, IaaS is IaaS, and containers are containers, essentially, whether you run it on AWS, Azure or Google Cloud.

SaaS and PaaS is where the differentiation in commercial cloud actually happens. For a company like Microsoft, its SaaS, such as Office 365, PowerBI, Dynamics, SharePoint, Teams, and Skype for Business are things that set it apart from the rest of the industry. These are application platforms that already have significant market share with on-premises, so moving customers to cloud-based, hosted versions of these as SaaS is a natural business for it to transition to from its legacy installations.

These workloads are already very sticky because they are tied into Microsoft's Active Directory authentication mechanism, which is a foundational technology for Microsoft-based environments. These customers are already locked-in, but it's not like they are looking to move off of these application platforms anyway, because there really aren't many good alternatives to them.

With Platform as a Service, you have all kinds of finished application services such as hosted databases and machine learning systems that are billed on a transactional basis. These systems, when combined with container-based PaaS, allow enterprise customers to build highly scalable systems that would otherwise be cost prohibitive to implement in IaaS and can be provisioned on-demand.

Also: SaaS, PaaS, and IaaS: Understand the differences https://www.zdnet.com/article/saas-paas-and-iaas-understand-the-differences/

One such example is DocumentDB https://aws.amazon.com/blogs/aws/new-amazon-documentdb-with-mongodb-compatibility-fast-scalable-and-highly-available/, a hosted database service that is API-compatible with MongoDB but does not use any actual MongoDB code, which Amazon launched this week on AWS. https://www.zdnet.com/article/aws-launches-fully-managed-document-database-service/

For now, one can build applications in AWS, using IaaS and container-based systems and back-end them into DocumentDB, and at a later date, move them back into on-premises or even another competing hyperscaler cloud such as Microsoft Azure or Google Cloud Platform. But this may not be the case indefinitely.

Today, many of these hosted services use APIs that are compatible with their open-source counterparts. So, the code is portable; it's not stuck at that cloud provider.

This is not entirely different from say, the classic issue of porting from one SQL-based database to another, as long as they are coded to ANSI SQL specifications. At that level of compatibility, it doesn't matter if a database started out on Oracle, you can then move it to IBM DB2 or even Microsoft SQL Server.

But as these services become commoditized, much like IaaS did for compute and storage, cloud providers will add their own feature enhancements, which will inevitably diverge from the open source counterparts. And software developers love to take advantage of new features, especially if they can increase performance, improve scalability, and can save money on transactional or computational costs.

That's one of the reasons why they are moving to PaaS and containerization and microservices in the cloud in the first place — to build true "Born in the cloud" apps. Plus, they can focus on running an application platform and their code instead of worrying about the underlying infrastructure. IaaS is really only just an intermediate step to the cloud for transforming workloads, because you still have to worry about maintaining the operating system stack.

Also: IBM: Still failing after all these years https://www.zdnet.com/article/ibm-still-failing-after-all-these-years/

But, as I often encountered at customers when I was at IBM in the mid-2000s, if you end up putting business logic into stored procedures and triggers on one particular database platform in order to take advantage of performance optimizations, you can end up having major compatibility headaches.

Then it's not so easy to move from, say, Oracle to IBM DB2. It could end up costing you a lot of software development time (and money) to move that business logic out of the database, so you can port it from one platform to another.

I remember the mainframe customer had 800 stored procedures and triggers in Oracle, and it would have cost them millions to remove all those and move the logic into middleware on J2EE instead. Although DB2 would have been cheaper than Oracle in terms of licensing, the software development costs would have been much more expensive. They ended up just sticking with Oracle, but moving it to a different operating system and hardware platform (IBM's AIX and POWER) to get the performance they needed. They were locked in to that database.

We could very well see this happen with hyperscale clouds. Sure, DocumentDB is MongoDB compatible now. But who is to say, five years from now, the new APIs are going to be identical? And DocumentDB is just one cloud service. A highly scalable, born-in-the-cloud application might be designed to take advantage of a dozen or more types of cloud services that are specific to that cloud provider. All of which are constantly evolving and getting new feature sets.

So you can certainly design cloud-based systems that are fairly self-contained and portable. But it may not be financially viable to do that long-term. The finished cloud services will be cheaper than what you can host in IaaS VMs or even in containers. With PaaS, the trade-off is ultimately going to be performance, features, and cost versus portability.

Is cloud lock-in an inevitability as we become more dependent on finished cloud services? Talk Back and Let Me Know.

Also: Top cloud providers 2018: How AWS, Microsoft, Google https://www.zdnet.com/article/top-cloud-providers-2018-how-aws-microsoft-google-ibm-oracle-alibaba-stack-up/

The more you lose control of the infrastructure and move your focus on strictly running your application code and having to depend on a hosting platform, the greater chance there is of that platform becoming sticky — which is precisely what hyperscale providers like AWS and Microsoft want. They want you to stay. They want you to continue buying cycles and transactions. They don't want you to move off their clouds. SaaS systems like Office 365, Workday, and Salesforce are obviously the ones that are sticky.

This is really no different from having on-premises software platforms, which are proprietary and use code that cannot easily be ported to another platform. The difference is that, instead of licensing these platforms, you are renting time on them, which the bean counters at your organization prefer, because it's an operational expense (OPEX), not a capital expense (CAPEX).

CES 2019: Weird and wonderful tech, gadg... (/pictures/ces-2019-weird-and-wonderful-tech-gadgets/)

PREVIOUS AND RELATED COVERAGE:

What a hybrid cloud is in the 'multi-cloud era,' and why you may already have one https://www.zdnet.com/article/what-a-hybrid-cloud-is-in-the-multi-cloud-era-and-why-you-may-already-have-one/

Now that the services used by an enterprise and provided to its customers may be hosted on servers in the public cloud or on-premises, maybe "hybrid cloud" isn't an architecture any more. What may the case, that's not stopping some in the digital transformation business from proclaiming it a way of work unto itself.

Cloud computing will virtually replace traditional data centers https://www.zdnet.com/article/cloud-computing-will-virtually-replace-traditional-data-centers-within-three-years/

Cloud data center traffic will represent 95 percent of total data center traffic by 2021.

What is cloud computing? Everything you need to know https://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-from-public-and-private-cloud-to-software-as-a-service/

An introduction to cloud computing right from the basics up to IaaS and PaaS, hybrid, public, and private cloud, AWS and Azure.

Cloud computing: Here comes a major tipping point https://www.zdnet.com/article/the-relentless-march-of-cloud-computing-continues-and-here-comes-the-tipping-point/

Application spending has moved to the cloud fastest, but other areas of IT spending are catching up.

RELATED STORIES:

- Microsoft Azure: Everything you need to know https://www.zdnet.com/article/microsoft-azure-everything-you-need-to-know/
- Inside Microsoft's Cosmos DB https://www.zdnet.com/article/inside-microsoft-cosmos-db/
- How Amazon Web Services runs security at a global scale https://www.zdnet.com/article/how-amazon-web-services-runs-security-at-a-global-scale/
- Watch out MongoDB AWS launches fully-managed document database service https://www.zdnet.com/article/aws-launches-fully-managed-document-database-service/

LOG IN TO COMMENT | Community Guidelines

Join Discussion

ADD YOUR COMMENT