I'm all 'bout that PaaS, no trouble.

Published on November 22, 2016

Let's use a practical example and look at how things like PaaS, DBaaS and blobs allow you to focus on building great web applications, while the cloud takes care of all the boring and complicated stuff. Don't exactly know what these acronyms mean? Great - this article is for you!

So you've heard that everyone is 'moving to the cloud' to do things faster, better, cheaper. But how does that work, and what exactly does it look like, what benefits would you be getting when you chose Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and/or other services instead? Here's a *simplified* use case with a made up company to illustrate it!

Service (PaaS) and/or other services instead? Here's a *simplified* use case with a made up company to illustrate it!

"Kytten.IO is a fantastic website for people who love cats, and even more love to share their photos and experiences with other people in the community. The website is growing in popularity which represents a challenge for the team and existing infrastructure. How can they leverage the wonders of cloud computing, without having to rewrite the entire application?"

At the moment, Kytten.IO are using a dedicated server sitting at a co-location space, and they're running Apache with PHP, with Wordpress as the engine to power the main components of their website, backed by a SQL database.

Scenario 1: Infrastructure as a Service (IaaS)

Server (hosting)   VM ('in the cloud')
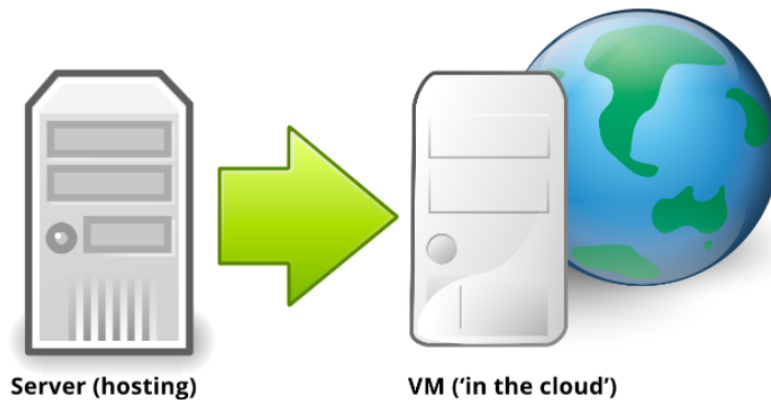
The company signs up for a cloud service that offers virtual machines and chooses one with similar specs that also comes with their choice of the operating system. They install & configure it the way they like, ultimately copy all code, database exports and files over and after a DNS change the company can retire their old physical server at the co-location provider.

Which aspects does the cloud take care of in this scenario?

Application Code	
Middleware (Config, Scaling)	⊖
Database (Config, Maintenance, Backup)	⊖
Media Files (Storage, Backup, Delivery)	⊖
Operating System (Config, Maintenance)	⊖
Server Hardware (Power, Cooling, Maintenance)	✅
Network (Uptime, Bandwidth)	✅

So, of course, everything is now *coming from the cloud*, but most of the things in their stack is still the same and the team from Kytten.IO still have to manage that. Will this solution really address their scalability issues? They could, of course, use virtual machines with more RAM or larger CPUs - or even more than one machine, but this is where things get tricky, and they're really looking for a way to reduce complexity and focus on writing their application & adding new features instead. Let's look at what else the cloud has to offer!

Blobs - Scalable Object Storage

Every photo that you see on Facebook, every song you listen to on Spotify is kept in object storage, which is a very direct way for applications to store and retrieve unstructured data. Think of it as something similar to a Dropbox kind of thing that your application can use, and you leave it completely up to that service how they do

application can use, and you leave it completely up to that service how they do scalability and backups - all you care about is dumping files there and embedding them in your application.

Doing something like that takes a significant load off the main application server - without the need to change much! For our solution, we could propose Kytten.IO use an Object Storage Service to take care of all static files like CSS, images, and videos. In their case, a simple Wordpress plugin would re-route every storage request to Object Storage. Read more about this concept and try it for free on Azure.

DBaaS - Elastic, managed databases

This should be an easy one. Of course, the company could spin up a separate virtual machine with a database on it to keep things separate and scale better, but they would still have to manage that other server, watch tablespaces, take care of backups etc. Database-as-a-Service gives you most of the functionality you're expecting from a SQL database, but it abstracts that complexity so you can just focus on writing smart SQL queries and not having to worry about database administration. Read more about this and try it for free on Azure.

PaaS - System-Administrator-as-a-Service

This is where things get really interesting. Remember when we spoke about the team configuring and installing stuff inside the virtual machine? It might be relatively easy - but do developers really know - really want to know - how many gigabytes of RAM the machine should have? How you patch operating systems or databases for vulnerabilities? How middleware scales? What if it was possible to have this abstracted as well and automatically taken care of? What if you could have a middleware *service* you could deploy your code to and it does everything else for you? At some point, the cloud could *decide* to run five instances of the application for you, and then three at some other time. Your 'servers' could crash and nobody would care, because the cloud has spun up a fresh one for you? Read more about this and then try it on Azure for free!

Scenario 2: Putting it all together

Server (hosting)   Cloud Solution
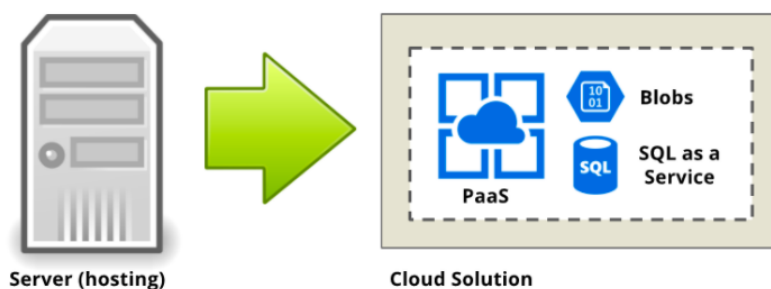
Even if we chose only one solution of the above, Kytten.IO would already have gained substantial benefit by automating routine & maintenance work and taking away load from the application server. They can do more with less! Now, let's combine all of these together. Static files are stored & served in blobs, the database is fully managed & elastic and the application is delivered through PaaS. In this particular case, Kytten.IO can even choose a Wordpress image and spin that up, set scaling rules and a lot more. The users of the Kytten.IO website will now be able to enjoy a great user experience and the company can fully focus on making the application better, manage the community and keep those kitten pictures coming!

Which aspects does the cloud take care of in this scenario?

Application Code	
Middleware (Config, Scaling)	✅
Database (Config, Maintenance, Backup)	✅
Media Files (Storage, Backup, Delivery)	✅
Operating System (Config, Maintenance)	✅
Server Hardware (Power, Cooling, Maintenance)	✅
Network (Uptime, Bandwidth)	✅

Once more -- this is a simplified example to illustrate the concepts. Let's talk about microservices and containers next!

21 Likes

1 Comment

Steven Chin   1y ...
🔹SME Cloud Transformation.Digital Modern Workplace. Director at UCloud Asia.Microsoft P-Marketer.