

The trade-offs for gaining a good multi-cloud strategy. An opinion paper by Philip Smith, CTO.

Back in October 2019, I wrote an article on LinkedIn about being able to leverage multi-cloud for compute.

My opinions on leveraging multiple clouds have not changed, other than that they clearly did not create a multi-vendor network between them and instead customers leverage 'Carrier Neutral Data Centres (or Facilities)'.

Let me just clarify that when I talk about a cloud(s) I mean AWS, Microsoft Azure and Google Cloud Platform (GCP). Whilst there are others, most notably IBM Bluemix and Oracle Cloud, they are not really comparable with the top three.

What is interesting about the top three is they all score differently across different Magic Quadrants, meaning you really should pick and choose the best for a particular workload.

There are two contexts when it comes to multi-cloud:

- 1 When we try to use multiple clouds to achieve the same outcome technologically (think workload mobility)
- 2 When we select the best for a product stack and house it in that cloud (think right tool for the right job).

The former strategy is meant to allow for portability across clouds for workloads. The primary drivers are usually avoiding vendor lock in and regulatory directives regarding resiliency and fear. In response to these, organisations that ECS work with tend to envision using multiple clouds as a way to prevent the impact of outcomes like a single cloud ceasing to trade, or suffering an outage, or even worse in some cases, a sudden increase in pricing.

However, in practice there are many architectural trade-offs that must be made to be completely portable across multiple clouds. Before I can discuss the real value of a good multi-cloud strategy, I need to point out that achieving this value relies on an Operational Model that sees the cloud for what it is; Software.

Using the right tool for the job

People, process and technology must ALL transform to become DevOps aligned.

Under this umbrella, frameworks such as Agile and strategies such as Infrastructure-as-Code and automation make the cloud accessible. People need training and coaching in all of these and more, otherwise the risk of building out cloud environments – as you would a traditional on-premise – are guaranteed to impact the value gained.

In the same way you would approach networking, I would recommend pulling together a team who have knowledge in multiple cloud technologies, but also recognise that each engineer will likely be a specialist in only one.

Take the similarities between Cisco and AWS:

Cisco has a very mature learning path – you would learn your basics and then you can do your professional level and so on and so forth. If you want(ed) to learn about networking, you went to Cisco and this gave you the grounding that you would need to be able to operate across Juniper, 3Com, HP or any other networking platform.

The same can be said for AWS; having been the first to the cloud market at scale. They have a set of mature tools and learnings that you can progress through and this helps you learn the foundations of cloud. But this isn't always enough.

One of the first challenges when considering your cloud strategy is the depth of knowledge engineering has over any particular cloud.

Remember how Cisco taught people to do networking? Very rarely did you meet an engineer who was an expert in multiple network vendor equipment configuration(s) at a detailed level. Sure, in some of the projects I was engaged in, having been Cisco qualified I was able to carry out basic configuration of a Juniper or 3Com, but never with the same level of expertise as I could with Cisco.

This is a key point. If you try and adopt a multi-cloud strategy in your enterprise, your engineers need to be able to specialise in at least one cloud, but they can't achieve mastery across more than one. For example, you can pass all of the exams across the major three clouds, but how much experience will this person have at the level required to really gain the value within the context of your organisation?

My dad once told me to "use the right tool for the right job". Whilst the context of that initial teaching came after I used one of his prized soft steel wood chisels to tighten a screw, I believe it remains relevant when we talk about multi-cloud strategies.

I would advocate assessing which is the right cloud to use for a particular workload (the entire stack). I would also try not to force that workload to be portable by selecting less than favourable technologies or trading out value. For instance, Docker and Kubernetes are a great way to make application workloads portable, but this only really works if the application is developed for containers. You really can't make a serverless application portable across multiple clouds. Additionally, just hacking existing entire systems (based on Linux/Windows) into containers isn't a great strategy either.

By setting a precedent in an organisation that you must use multiple clouds to make workloads portable, you write off many of the advancements that you can gain from specialising in one particular cloud. For example, the Serverless Application Model (SAM) in AWS requires the use of AWS specific services and these really are not transferable as you're using the smallest units of configuration/code already. Successful usage of the SAM requires deep knowledge around how to configure and control supporting infrastructure such as DynamoDB or API Gateway. Each of the major clouds that we are discussing here have services that do similar things, however, how you configure and control them is often different. How you operate them will also be different, in terms of what you are meant to measure and how you respond to issues.

Ultimately large cloud players are providing value on top of the ability to be able to run workloads on their cloud and it's these additional features that you can use if you specialise in a single cloud. Alternatively, if you try and operate across multiple clouds to gain portability, it's unlikely that you will be able to achieve the best in class applications across the best cloud, again bringing us back to the right tool for the right job.

The cost of achieving a good multi-cloud platform

To use the cloud correctly, you should be looking at adopting the best practices for each of those clouds. But be warned, looking at the identity and access management services across all three clouds requires a serious amount of study and implementation.



There are obviously nuances to each cloud and being able to have a single expert that can operate across all three at a detailed level is probably not realistic. An organisation that is going to operate with a multi-cloud strategy will need to understand the two, or three different clouds at a very, very deep level and therefore engineers need to be given the time to study the cloud that they're going to be operating on. It would be wise to have them as an expert in one and then be able to manoeuvre in the other clouds that you're going to support. This does mean however, that it is likely you will need to duplicate resources on a per cloud basis.

This is where Center of Excellence (CoE) or Guild based groups can help shape the usage – but this should only be to support harmonising the methodologies (best practices, principles, guardrails) for cloud consumption, rather than to trying to standardise the application architectures and configuration processes.

If you're using multi-cloud (for value and not portability), you can architect services working closely with your developers and engineers, who should be able to make decisions on which cloud offers the best services for a particular workload. The factors that will influence whether you host on a single cloud would be tied to the velocity with which you can continually add value to the service AND the compatibility with the features required by customers. So, for example, if each cloud has higher level AI services, you may choose GCP because you want to enable your applications with Natural Language Understanding – in my opinion, Google would potentially be the best cloud for this. You may even have a pre-existing application that you will need to re-host/factor into the cloud chosen and whilst this may influence the decision, it should not be the major factor.

If it is a business-critical application that you're intending on building out onto multiple clouds for portability (at the trade-off of realising deeper level value from single cloud) then the organisation needs to understand which parts can be portable. This often involves breaking down applications into microservices that can offer a way for you to better establish portability. To access a true representation of the trade-offs here however, you would need an experienced architect for each cloud to review what is lost by maintaining portability.

Value vs portability – the trade-off

As time goes by and greater dependence is placed on the strategy, more and more applications are either migrated to or built on top of cloud. This is where the impact of the trade-off is felt. The two outcomes will both have negatives as perceived by the enterprise;

- 1** If you decide to leverage multi-cloud for portability, you end up with knowledgeable engineering teams that are generally naive to the benefits available in each cloud. Their focus becomes mastery of the intermediary tooling selected to manage portability, such as Container Platforms, Configuration Management or Infrastructure-as-Code.
- 2** If you decide to leverage multi-cloud because it enables you to select the best platform for your service, then you will need a larger number of teams containing more deeply experienced engineers. They make choices about the tooling they use to drive the right cloud based on the desired outcomes for each service, which means they focus less on the intermediate tooling and more on the cloud itself.

Additionally, the layer of security, governance and controls also influence the usage of these tools and strategies and this adds to the complexity of consuming the cloud – however to the organisation, operationally they have 'saved costs' by limiting accessibility to cloud through common formats. The real impact will be that they lose velocity (due to additional complexity) and potential value, by being prevented from leveraging ALL of the services.

For this model to work, there needs to be an intermediary such as the CoE or Guild that can ensure architecture across multiple clouds, via individual teams, is aligned to an organisation's governance and strategic models. Complexity here is actually in running the end result, with tooling and processes needed to harmonise monitoring and alerting strategies, governance, technical controls and cost reporting.



In short, multi-cloud done well becomes a vastly more expensive effort.

Let's consider a real-world example. When you bring together your teams in Governance, Security and Controls and two of the services that form your application estate both use different clouds, you will end up with a set of controls that are to be applied to the underlying cloud, with the intent of maintaining compliance and protection of customers. These controls would need to be implemented by the intermediary configuration tool (most likely). For the sake of an example, let's say Terraform.

Deeper still, if you are using standard compute (an instance with storage) your organisation will probably have a control around ensuring that the disks are encrypted. This control cannot be implemented with the same code across multiple clouds and therefore when creating a control that ensures the disks are encrypted, you would need to use different code. You could use the same tool to apply the code, Terraform for example, but the code will still be different.

Here is the example in Terraform if you are interested:

- AWS applies encryption to an EBS volume, which is then attached to an instance. This offers independence from the instance and the ability to snapshot/move the disks
- Azure applies encryption to Managed Disks, in a similar way to AWS
- GCP again in a similar way

If you are not technical, I can tell you (and you will have to believe me), these are not the same feature structures, even though they are all designed to achieve encrypting a disk. Whilst they use the same tool to apply to the relevant cloud, you couldn't just copy and paste the specification and change the cloud provider. Ultimately, what you need will be three different specifications, probably backed by expertise capable of managing the differences across three different clouds.

Another major reason cited for using a multi-cloud strategy beyond service mobility is the perception of being tied to a supplier. This may come as a response to years of being tied into organisations like Cisco or Oracle, and there is a definite risk that being tied into a single cloud could have an impact if that cloud ever did increase its pricing.

The evidence however is not pointing towards this being the case. Instead, services are constantly being upgraded as you would expect with SaaS platforms. This means new value is being added to the services daily/weekly/monthly and instead, what we have seen is price reduction from the start.

As an example, Amazon Web Services started in 2006. After 14 years, we've seen an enormous number of price reductions in each service, and a massive increase in the types of services that are provided – including additional features in each of those services. Whilst many organisations have contributed to the evidence that third parties can tie you in and then charge you extra, that's not really the case with Amazon Web Services, and this goes for Google and Azure as well.



Closing thoughts

This opinion paper is not intended to criticise a multi-cloud approach, it is intended to highlight the difficulties and additional investment required to succeed in reaching an organisation's goals. If you or your organisation are considering leveraging multiple clouds, consider the cost and impact of doing so well.

The reality is, if you architect your services that are on cloud in line with the best practices published, you're more likely to avoid outage, unpredicted cost increases (unless through unexpected customer interaction, or poor usage of cloud services), and fines for breaking out of contractual obligations early.

Here are some thoughts you can consider:

- 1 Master one cloud before attempting to master multiple clouds
- 2 Test theories quickly in regards to cloud platform selection
- 3 Provide adequate support to teams involved with onboarding cloud platforms
- 4 Prefer velocity, value to the customer and mastery over fear
- 5 Leverage vision, mission and goals to achieve alignment to standards

More about the author: Philip Smith is the CTO at ECS. Philip has over 20 years' experience in the IT sector and is currently leading ECS in its Technical Capabilities and setting the strategy for increasing value for our client's customers. He is achieving this by understanding enterprise Cloud transformation pain points to help shape the solutions ECS delivers, providing strategies for customers to unblock stalled transformation programs and implementing strategic partnerships to enable ECS and client growth and success.



Why we're different

Our consultants deliver cloud, customer experience transformation, continuous improvement and engineering capabilities in one team. This means you get process, engineering and best of breed technology to deliver better customer experiences. Our experience working with some of the largest and most heavily regulated businesses in the UK to accelerate their customer journey transformation also means we put compliance and security at the heart of every solution.

If you are looking for advice about whether your enterprise would benefit best from a single or multi-cloud strategy, contact us today.

ecs.co.uk | 0207 403 0477

enquiries@ecs.co.uk