



GUIDE

10-STEP GUIDE TO THE COST- EFFECTIVE CLOUD

The cloud can serve as a sensible option for those looking to maintain their applications off-premise, but only if the costs make sense. While it can often save an organization money in support and maintenance, that's not always the case.

Before embarking on a journey to the cloud, every company should first consider how to leverage the cloud in such a way that's cost-effective.

Stuart Williams

Magenic Cloud Practice Lead Consultant

August 2018



Preface

Before we dive into the steps for conserving cloud costs, it's important to address a very important question: what is the cloud? The cloud is a general-purpose computing and data platform that exposes virtually unlimited resources. Traditionally, public clouds such as Microsoft Azure, Amazon Web Services, and Google Cloud Platform are the ones people think of. Although large-sized private clouds built on OpenStack or VMWare vSphere and supporting infrastructure are also clouds.

Specialized public clouds also exist, such as Salesforce and Pivotal.IO. Then there are the new container clouds that can be hosted atop any of these such as Docker, Kubernetes, and Pivotal Cloud Foundry.

Other criteria that make an offering a cloud:

- Supports Windows and Linux workloads and generally a large list of programming languages and platforms
- Has different models of cloud hosting: IaaS, PaaS, & SaaS
- A mature cloud has all the typical enterprise building blocks and patterns, plus an enormous catalog of 3rd party components

For the purposes of this guide, the advice given generally applies to the following cloud platforms:



Here are 10 steps or strategies to consider.

1. Know How Much You Spend

It may seem obvious, but having a firm grasp on how much you're investing in the cloud is imperative. Review or monitor your costs using the vendor's cloud management web portal and reporting tools; be on the lookout for changes in spend, month over month. Measuring too finely will generally not yield useful insights and can cause teams to spend cycles investigating what could be transient or event driven utilization. Managing spend on the cloud is a marathon not a sprint.

Consider investing in a 3rd party cost management tool especially if you have a multi-cloud strategy, have a large cloud presence, and/or have cloud centers in multiple countries. These tools can do a good job of rolling up and normalizing spend, and allow drilling down into cost areas for more nuanced analysis.



"You can't manage what you don't measure."

– Peter Drucker





For your IT team, make sure to segregate your tenants, business units, and/or environments to more easily track spend in categories. Insist on a naming convention to make granular resource ownership (and usage) easier to track.

Here's what should be measured:

- Total cost of an application broken down by deployable unit or components (web site, web server, headless process, database, enterprise queue, etc.). For shared services, it is useful to be able to meter the use by each business unit or customer to properly allocate cost. For websites, additional useful information can be gleaned by instrumenting the web site with Google Analytics or a similar tool
- Storage consumption trends compared to your business growth, checked by any data-retention policies (if any)
- Periodicity of demand by application system and growth trend over time. In other words, if your system has increased usage during the weekday and less at night, and your system is scaled up and down accordingly, establish the overall trend by the low and high points in the scaling measured over time
- Volume and cost of ingress and egress of data in and out of your cloud centers. This is often an under-appreciated cost and tends to be price sensitive even by time of day. Comparing flow of data with the associated quality-of-service (QoS), can yield some interesting facts
- Growth in total spend by customer or business unit is also instructive. For systems that have associated revenue, comparing cost vs. usage vs. revenue by customer will inform the business about how the system is "value priced"

Pro Tip: *Not all hosting from the same vendor is of equal cost. For example, the cost of the exact same VM in two different data centers operated by the same vendor may vary significantly.*

2. Know Your Application Profiles

All application systems should have an expected profile as part of their metadata. For each of the deployable units, there should be profile data calling out the expected resource utilization of that component:

- CPU
- RAM
- DISK
- NETWORK

////////////////////
"Measurement without action is an opportunity missed."

– Anonymous





This data should be kept in a format so that is possible to compare multiple applications against each other to form a generic profile for that component type. Examples:

- Java Spring Boot Web Site
- .NET Core Web API
- Azure Function that services a queue
- Medium SQL Server instance (DB ~100 – 500 gb)

Application components that significantly exceed the profile elements should be investigated with tools and tuned accordingly.

Additionally, running applications should be profiled for performance and processing bottlenecks. These should be noted and added to the backlog for teams to address.

The movement of data in and out of a system is often a culprit driving cost. Here are some things to consider:

- Use the appropriate tools to identify and fix badly performing/costly data access
- Make sure your applications ask for the right amount of data
- Watch out for large but unnecessary data moves, especially across the cloud/network boundary

Pro Tip: *Make sure performance problems and excessive resource use are at the top of the application owners backlog as “must fix” items*

3. Don't Leave Unused Resources Running

First and foremost: some resources do not cost money when idle, while others do. Know which is which!

Turn off resources not in use and turn them on again when needed. Other cost-saving practices include:

- Shutdown VMs automatically on off hours
- For a resource that cannot be turned off, script its creation/destruction so it can be dynamically provisioned
- Delete staging slots when you are sure the deployment is okay



“Last one out, turn off the lights.”

– Anonymous





4. Be at the Right Scale

Knowing when to ramp up resources and, conversely, ramp down can make a significant impact on your bottom line. The ability to strategically scale is often overlooked and frequently a source of wasted costs. Tips regarding scale:

- Use auto-scaling to deal with burst situations and to reduce costs during off-hours
- Size IaaS resources appropriately and measure their utilization for optimal spend
- Carefully consider how many compute and scale units you need vs. how many you are using

Pro Tip: *Can you use cheaper storage? Most cloud solutions offer various levels of storage at different price points.*

////////////////////////////////////
“Scale is not a constant; measure and adjust.”

– Magenic Cloud Team

5. Consolidate

Consolidating can be a useful strategy when done with care. The art is to balance consolidation against performance and availability. Incremental consolidations, followed by a period of monitoring to make sure the systems involved are not negatively impacted is key to success. The rationale being, by sharing resources you can save money if you are careful. However, don't consolidate purely usage-based resources unless there is another compelling reason.

////////////////////////////////////
“Consolidate then measure to confirm expected gains.”

– Anonymous

Consolidating is a natural move for certain deployable unit types:

- Web Applications
- Web Services
- Databases

Anti-examples would be:

- Enterprise Queues
- Cloud Storage
- Steaming Analytics

These are all in the “taxi model.” For example, if you get billed per transaction, KB, or message, consolidation not only does not make sense, it can make management and cost allocation more difficult.

Pro Tip: *Vendors and 3rd parties offer services/software to advise on consolidation*





6. Don't Use Deprecated Features

Deprecated features are counterproductive to cost management because:

- They may be going out of vendor support or if support is available it generally comes at a premium price point
- If a vendor has deprecated a feature, chances are there are better alternatives in both price and performance
- They can expose security vulnerabilities and if these are exploited can result in significant cost in both dollars and reputation. The latter being the hardest recoup

Pro Tip: *Someone should be responsible to monitor vendor announcements and roadmaps and inform impacted stakeholders*



“Deprecated: soon unsupported. Unsupported means vulnerable.”

– Magenic Security

7. Migrate to PaaS

Moving workloads off VMs or deprecated facilities on to the equivalent PaaS services will generally have many benefits:

- Typically, lower cost
- Better business continuation options
- Superior scalability and higher availability options
- Easier operability
- Potential to consolidate during or after migration to the cloud

In addition, there are often capabilities that are not available in the equivalent server product.

Pro Tip: *Don't be afraid of “vendor lock in.” There are usually alternatives from other vendors should you decide to switch providers. In other words, don't let fear of vendor lock-in prevent you from enjoying the cost savings, features, and other advantages of leveraging PaaS offerings. Open source and a wide variety of commercial products are available on many cloud platforms and give the business flexibility.*



“PaaS offers the most leverage, features, and flexibility for cloud solutions.”

– S. Williams





8. Migrate to Containers

Containers are not (yet) a replacement for all workload types, especially enterprise persistence, but they are excellent places to put compute resources such as:

- Web Site and Services
- Headless Workloads: Tasks, Jobs, and Event Processors

Generally, a workload is a good candidate for containers if:

- It is stateless
- It supports a re-start able unit of work
- The units of work are short (measured in milliseconds, not hours)

Containers, when useful, generally have much better densities than VMs (which were a considerable improvement over dedicated servers). In addition to saving money, they can also be deployed, scaled, moved, and managed easier, each of which can have cost avoidance if not flat-out savings in addition to the benefits of PaaS listed above.

At Magenic, we have had good customer success with Pivotal Cloud Foundry (PCF) and Kubernetes (K8s) on a variety of public and private cloud.

Pro Tip: *Make sure that the workload is suitable for containers before moving it. Pull-the-plug tests are the best way to make sure of this.*

9. Consider Caching

Caching helps avoid repetitive data reads, but it comes at a cost of adding complexity and can induce consistency issues. Building caching into your apps with REDIS (or similar functionality) can be a significant performance and cost improvement, especially for data that is short lived.

Can your app benefit from a Content Delivery Network (CDN)? CDNs are ideal for static resources that change infrequently but are used by many as a way of offloading the request to the server to the cloud. Frequently loaded resources but infrequently changed things benefit from CDNs, such as:

- Images
- Media files
- Shared JavaScript Libraries
- Cascading stylesheets (CSS)
- Etc.



“Our observation is that the best density of compute resources comes from using containerized workloads.”

– Magenic IT



“There are only two hard things in computer science: cache invalidation and naming things.”

– Phil Karlton





Can your web app cache pages at the web server or browser to avoid re-invoking the business application logic?

Examples:

- An API that exposes lookup data that almost never changes (like zip codes), can add HTTP caching directives that cause the web server or browser to hold onto the results for the desired period
- Images and other static content (not on a CDN) can benefit from this approach
- Single Page Applications (SPA) or Progressive Web Applications (PWA) put significant functionality on the client from the server and off-load work from the web server

Pro Tip: *The caveat is that when the data does change the cache gets invalidated consistently across instances lest odd behavior occur. Test, always, and often.*

10. Leverage Your Vendor Contract(s)

Customers who have longer term contracts generally get:

- Better Pricing
- Dev/Test Environment Discounts
- Credits for R&D and Pilot Programs

Cumulatively, these can result in significant savings.

Try to manage your vendor relationship for the long-term (at least three years), and make it clear what your expectations for savings, as well as any concerns over security, performance, and service level agreements (SLAs). Get these agreements in writing, and measure them to make sure goals are being met by both partners.

Take advantage of program funding with partners. Vendors often have special incentives/ funding you can leverage when working with a partner such as Magenic to offset the cost of these scenarios:

- Competitive re-platforming
- Adopting new technologies
- Moving to the cloud
- Early adopter scenarios

Pro Tip: *Talk to Magenic about how we can help.*



*"Price is what you pay,
value is what you get."*

– Warren Buffet





Conclusion

No matter where you land on your migration to the cloud, it's important to know how to leverage it in the most cost-efficient way. The steps listed in this guide help identify ways to conserve spending, but navigating the cloud journey can be a complex and difficult endeavor. The Magenic cloud practice can ease that burden for your organization. We specialize in helping companies migrate and modernize to the cloud, not just for cost savings, but for the sake of increased business agility and speed to market.



About Magenic

Magenic is the digital technology consulting company built for speed. We have the right digital strategies, the right process, and the right people to get our clients digital products to market faster.

Visit us at magenic.com or call us at **877.277.1044** to learn more or to engage Magenic today.



This document is for informational purposes only. Magenic Technologies, Inc., makes no warranties, express or implied, in this summary. Other product and company names mentioned herein might be the trademarks of their respective owners. © 2018 Magenic Technologies Inc. All rights reserved.

