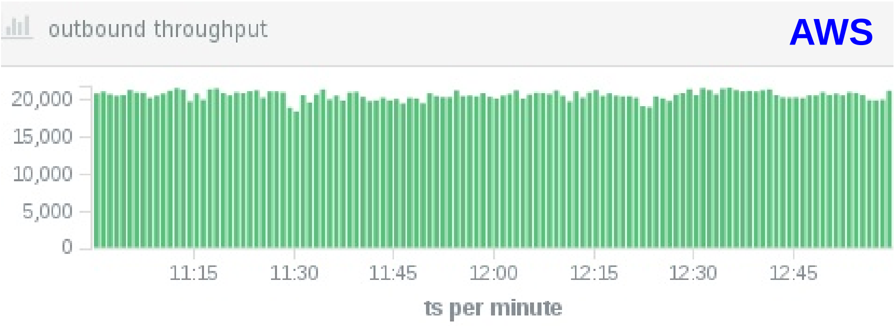
Clouds Compared

How a polyglot persistence open source micro-service performed differently when run in Amazon's, Google's, and Microsoft's cloud.

by Glenn Engstrand

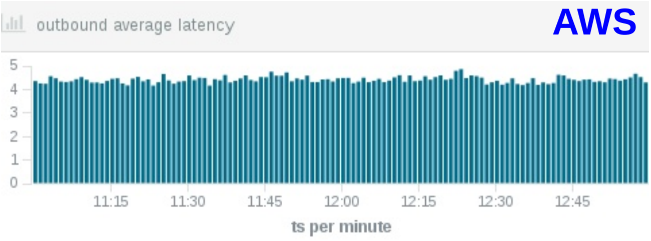
We all know that, in the world of cloud computing, Amazon Web Services is the leader by far.



Network intelligence group, Synergy Research, recently released a report revealing that cloud vendors Microsoft, IBM, and Google had increased their market share by 5% (combined). I thought that perhaps it was time to see if there really was any viable alternatives to AWS in 2017.

What has turned me off to Windows Azure in the past was that their only PaaS relational database offering was MS Sql Server. Recently, they announced a preview version of MySql. This is what prompted me to include them in this evaluation.

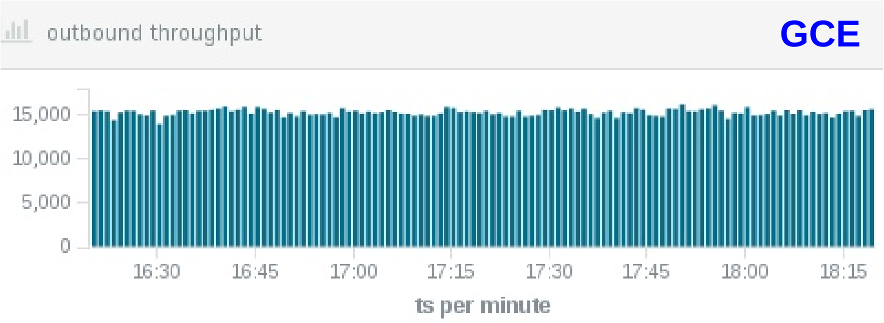
I decided to conduct an experiment. I ran my already developed news feed micro-service load test on the Amazon, Google, and Microsoft clouds then compared them in terms of User eXperience, price, and performance.



The accounts that I used to conduct these tests had no special status associated with them. To Amazon, Google, and Microsoft I was just some guy with a credit card.

For all three vendors, I used their web application instead of the command line interface. Like most of the online world, all three cloud offerings entice you with a freemium model. You can’t run these tests with the free part on any cloud so I also covered how each vendor carried out the upsell in their console experience.

Why bother with comparing upsell tactics you ask? Many readers most likely have a five or six figure monthly cloud budget and couldn’t care less about the free tier. There are two reasons for covering upsell. It is a natural roadblock so how they handle upsell is a good predictor for UX in general. Managing operational costs is even more critical at web scale. I believe that it is prudent, when evaluating a cloud vendor, to consider what tactics that they would be capable of in order to raise your bill.



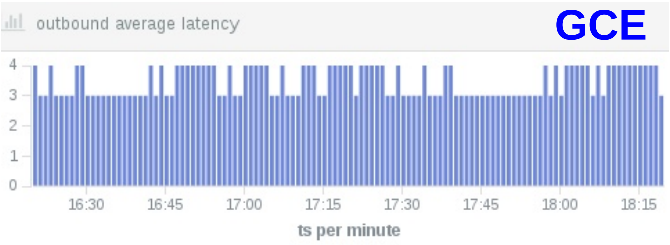
The news feed load test consists of running the DropWizard version of the micro-service and all of its dependent services on 6 CentOS instances and a MySql instance. I run the load test application for at least a couple of hours and collect performance data in Elastic Search via my news-feed-performance application. I had to go with that approach instead of Kong because I could never get Kong to install properly on the Google Compute Engine. Because I used CentOS, I was able to use the same devops assets on all three clouds. The only tweak I had to make was to specify a user name of feed@mysql-feed when connecting to MySql on Azure

It may be unfair for me to compare UX because I am much more familiar with AWS than with the dashboards of the other two cloud vendors. I do tend to see everything in terms of how it compares with the AWS dashboard. Having said that, I would still claim that the Amazon experience is the superior one.

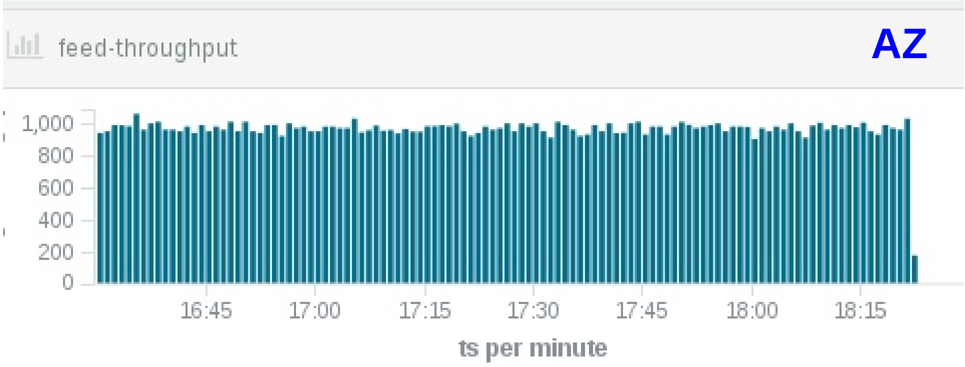
Everywhere in the AWS dashboard, you are always presented with the ability to configure and / or deploy more than one. Where that is most critical is in configuring and deploying compute instances.

Not so with the others where you have to fill out this multi-step funnel over and over again for each instance specifying the many options the same way. With GCE, some of the settings for the next instance would default to what you specified in the previous instance but not always and not predictably. AZ claimed to get around all this repetition with templates but I never could figure out how that worked.

Another nice detail with Amazon RDS is the ability to create the first user and database when provisioning the instance. With GCE and AZ, you had to use SQL and the MySql client command line utility. Not the end of the world but still.



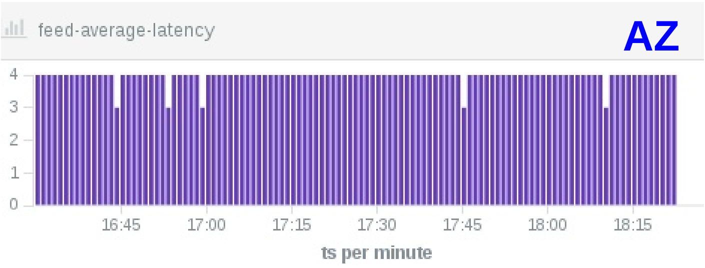
One of the first roadblocks to a cloud provisioning UX is configuring the firewall rules. Amazon gives you the ability to do that from everywhere that you would conceive of wanting to do that.



With both Microsoft and Google, you have to go to a specialized and not well surfaced area for configuring firewall rules. With AZ, firewall rules for compute instances have a very different GUI than for a database instance. It took me a while to figure out where to go to get to where GCE keeps its firewall rules configuration. Ironically, I had to Google for those instructions.

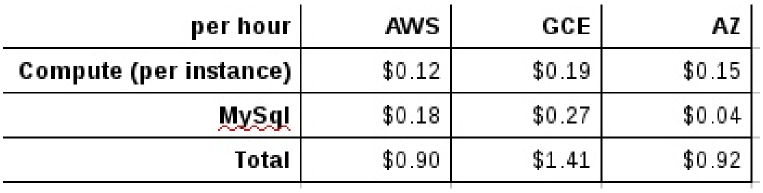
My first introduction to resource quotas was an error message.

The bad news about Amazon upsell is that you hit the paywall very quickly. It may be fine for first timers but you won't be able to use the free tier for any load based performance evaluation. The good news is that there are no roadblocks. Once you agree that you are not on the free tier, you are not restricted anymore.



The Google upsell had more impediments than the Amazon one. My first exposure to resource quotas was an error message that prevented me from spinning up the third instance. It took one email to clear and I had to deposit $35 in my account. I could still use the $300 sign-up credit (good for a year) so the upsell hasn't actually cost me anything yet.

Of the three cloud vendors, the Microsoft upsell broke the flow the most. Like with Google, my first exposure to resource quotas was an error message that prevented me from continuing my test. Unlike Google, it took two email messages, a subscription upgrade, two quota increase help support tickets, and a week to resolve. That first email was to a Microsoft representative who contacted me first. I never heard back from him again. On the dashboard, you had to request quota increases per instance type. I could still use the $200 sign-up credit (good for a month) as well.



per hour	AWS	GCE	AZ
Compute (per instance)	$0.12	$0.19	$0.15
MySql	$0.18	$0.27	$0.04
Total	$0.90	$1.41	$0.92

The Microsoft portal had the worst UX too. The site was very sluggish. It took a lot of interaction just to complete the most simple tasks. The information architecture was buzzword bingo deluxe. Most of the defaults made no sense. My ISP is in California so why would the default data center be in East Asia?

When compared to AWS, GCE was 50% more expensive but throughput was 75% and AZ was priced comparatively but throughput was only 5%.

We have covered UX but there is a more important basis to compare these three different cloud technologies; namely price and performance. Let’s get down to the actual load tests on the three clouds and compare the price and performance of those tests.

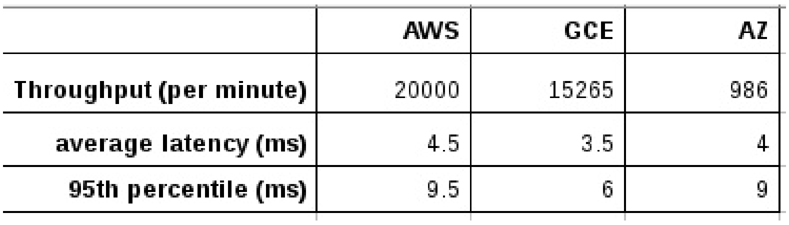
For AWS, I used m4.large instances in their Ohio data center. The relational database was MySql 5.6 on a db.m4.large with a 100 GB SSD.

There is a bandwidth cap of 450 MBPS on this type of instance. The performance results on AWS form the basis by which I compared the load tests run on the other two cloud stacks.

For my first test on GCE, I used n1-standard-2 instances in their Iowa data center.

The relational database was what they labeled as MySql second generation running on a db-n1-standard-2 with 100 GB SSD. I chose the n1-standard-2 instance type because the specs looked comparable to the m4.large in EC2. These GCE instances are one third cheaper than the EC2 instances but also half the throughput during the load test.

For my next test, I upgraded to using n1-standard-4 and db-n1-standard-4 which is supposed to be twice as powerful.



	AWS	GCE	AZ
Throughput (per minute)	20000	15265	986
average latency (ms)	4.5	3.5	4
95th percentile (ms)	9.5	6	9

These GCE instances are actually one and a half times more expensive than the EC2 instances. The throughput from the load test on the GCE instances was only three fourths that of the throughput on the EC2 test. GCE imposes a bandwidth cap of 8 GBPS on this type of instance.

The latency for all three clouds was very similar.

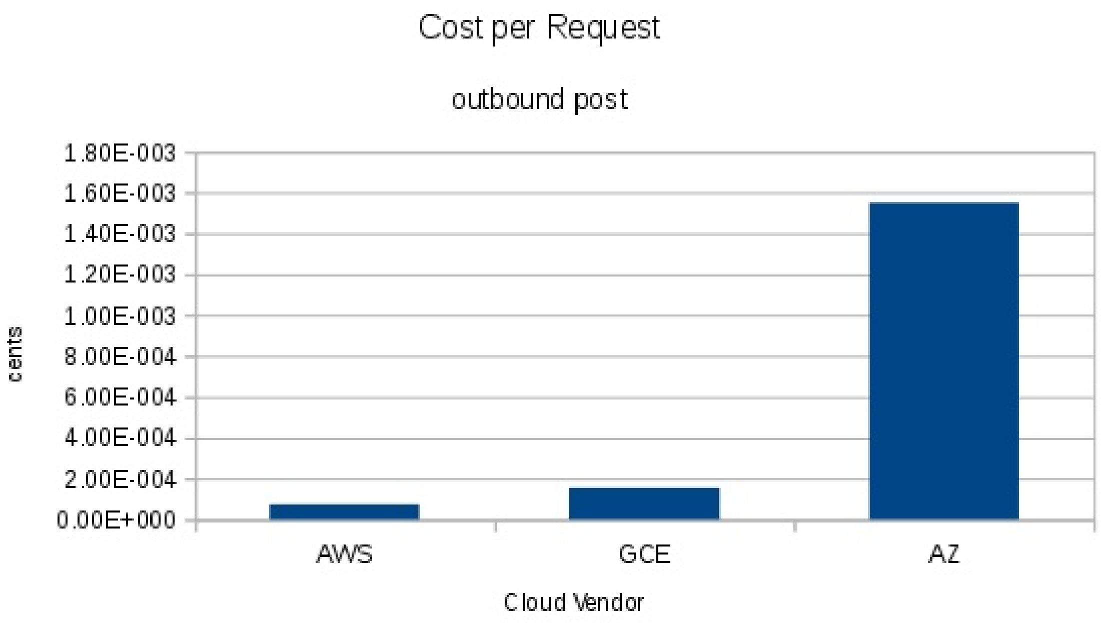
For AZ, I used DS2_V2 instances in their Virginia data center. The relational database was what they labeled as MySql basic on 100 compute units and 50 GB.

The GUI promised that standard and premium would become available eventually.

AZ costs were comparable to EC2. The throughput of the load on AZ was only 5% that of the throughput of the load on EC2. There is a 1 GBPS bandwidth cap on this instance type. I suspect that the throughput will improve with standard or premium versions of MySql. The question is will it improve by a factor of 20?

Latency was comparable across all three cloud vendors covered here.

Are there viable alternatives to AWS? For those looking to move an already existing application whose tech stack is similar to what was tested with here the answer, quite frankly, is no. At least not right now.



http://glennengstrand.info/cloud/performance/aws/gce/az