



The Cloud Chronicles: Unbiased straight talk from real cloud customers.

There is no shortage of opinion out there regarding the value of one cloud provider over another. We want to provide our 1,500+ cloud customers with a platform to share their story, unfiltered and unabridged, so that others may benefit from what they have learned along the way.

Part One: Why Google Cloud?

An interview with the VP of Technology Strategy at Birch Hill Equity.



Interviewer

Scott Thomson

Director, Business Development
Public Cloud
Softchoice



Guest

Michael Young

VP, Technology Strategy
Birch Hill Equity

“Almost everything Google Cloud does is serverless and that’s the thing that pushed us to Google.”

Multicloud is a clear reality of the next era in cloud computing. As such, we are seeing many of our customers revisiting their cloud strategy to leverage the distinct advantages of various cloud platforms while also balancing risk.

What I’d like to understand from you as VP of Technology Strategy at Birch Hill Equity, an organization that’s quite advanced in its cloud strategy – is your perspective on multicloud and why you chose Google Cloud.

Multicloud certainly ties into our strategy at Birch Hill, especially with the numerous portfolio companies that I’m involved with. Our number one priority is for as much of what we do as possible to be serverless. This is mainly because of the skill sets we have available to us.

We have some people who can write good scripts – complex things. But I don’t necessarily have anybody who would do site reliability engineering or redundant management or anything like that.

When I look at serverless concept in the broadest sense –from the simplest forms of lambda functions to more complex products like Databricks Serverless and Apache Spark—pretty much every choice we’ve made is to align with a serverless architecture pattern. Almost everything Google Cloud does is serverless and that’s the thing that pushed us to Google.

So, obviously serverless, at least in my understanding, has specific use cases. Are you running enterprise applications in the cloud or is your main focus on data-related workloads?

The applications we build at Birch Hill are data-related workloads. But we do have a custom enterprise application at one of our companies that we’re refactoring to be completely serverless. It does have a front end, but that is probably going to be a thin presentation layer. All the functionality in the actual application itself will be refactored as serverless.

You wouldn’t run SAP if you were running an enterprise application. With serverless, you have to have an actual infrastructure. When it comes to custom-built things, that’s probably the better way. But it’s hard to not be able to do things serverless.

“We wanted to focus our time on delivering analytics products, not maintaining our cloud.”

Can you tell us a little bit about your journey from a cloud vendor perspective? You were one of our first customers with AWS. I would like to have a deeper understanding of why you incorporated Google Cloud into your strategy?

Birch Hill started using AWS before my time here. The only things being used were a PostgreSQL, Databricks and a lightweight data warehouse. That’s pretty much everything that the data center was doing. There weren’t a lot of applications being built yet, except for a few images being run.

The primary usage for the data frame stream was PostgreSQL. That was across the whole of Birch Hill. How it was architected presented significant security issues, as it was a single node with separate schemas for each company. This could have led to a big data breach. If one of our companies had a breach in that node, then by default everyone would, so we needed to resolve that issue fast.

After considering different options like Amazon Redshift and Snowflake, we decided to migrate to Google BigQuery. This was because we didn’t have to worry about fine-tuning or any of the infrastructure behind it. Regardless of the size of the data – whether it’s a single gigabyte or four terabytes – it’s always sub-second response time without having to lift anything. We wanted to focus our time on delivering analytics products, not maintaining our cloud.

We had also looked at running Elasticsearch on AWS. But the security and managed services on Google were far better than other cloud vendors because all of it hinged on the Google identity in the background.

As soon as you set up a project, you have a logical network that defines all your service accounts for you. Secure access is immediately granted through OAuth (open authorization). We are a team of six people running some meaty analytics, so Google was by far the easiest choice for us.

From a previous conversation I recall you saying that for your particular workloads “... AWS was requiring more effort to manage and maintain.” Is it fair to say it’s different with Google?

One hundred percent. For example, at one of our portfolio companies, we’ve got one project that now has seventeen cloud functions, a couple of different PostgreSQL databases along with a data warehouse and one VM that’s going to be running soon. All that required zero IT whitelisting and zero service account setup because of all the defaults within Google. The maintenance on running the infrastructure is almost zero for us. We wanted to focus our time on writing the code versus managing the infrastructure. That’s another big plus for Google.

Yeah, absolutely. I sum it up as a consumer-like experience with Google Cloud. It follows the excellent user experience design in all Google’s products, making it incredibly intuitive.

“You’ll be far more productive on Google if you just want to launch a product fast.”

What’s next for you as it relates to cloud, or Google in particular?

As we are now getting more complex in our workloads, we have some heavy Spark jobs currently running on AWS. We’re using Databricks as a managed service on top of it. This effectively manages EC2 clusters for us. From that perspective, we still have a multicloud, but the issue right now is security token management. It’s not great to port JSONPs or P12Ps across multiple environments. It’s a challenge we are facing with managed services across multiple public clouds.

Google has an equivalent product to Databricks, called Dataproc. It’s a simple, cost-efficient and fully-managed cloud service for running Apache Spark and Hadoop clusters. But there’s a lot more you have to put up to automate scheduling. We have to make a choice between getting good at Apache Airflow within Google to have ephemeral clusters, or get really good at managing security across multiple clouds.

What is your biggest challenge moving forward?

Companies like ours that lack internal DevOps capability are businesses that don’t have infrastructure or just want to write their code and host it. We’re probably at a point where we need DevOps and we need good infrastructure architecture design skills. But it’s not our main priority to build that. If I have another headcount opened, I want to hire another analyst versus hiring an engineer.

Last question, if you had any advice for other businesses who are thinking about a multicloud strategy and Google Cloud, what would you tell them?

You’ll be far more productive on Google if you just want to launch a product fast. With App Engine and BigQuery, you can stand up a data warehouse in a matter of hours using software-as-a-service and ETL tools. They just snatch that data into BigQuery and regardless of the size, you’re going to get immediate response by going through Google.

Our focus is on delivering the end product, so we always go to Google first. But, if we need something with really specific infrastructure components, then our second choice would be going to AWS.

For example, one of our projects has a lambda function that runs in AWS. But it’s triggered by Google because when we wrote the lambda function, we needed a Docker container. We needed an OS-level component in this function that Google didn’t support.

If you want to focus on building and managing infrastructure, I would recommend AWS because Google wouldn’t be able to support that as of yet. But they’re closing that gap fairly quickly.