egen

10 Ways to Reduce Cloud Costs

AUGUST 2020

The global economy has drastically shifted due to COVID-19. In our new economic reality, it’s never been more important to increase focus on high-value projects, differentiate your solutions from competitors, and most urgently, find new ways to improve the bottom line.

By accelerating digital transformation efforts and being hyper-focused on IT cost structures, you’d be amazed at how much opportunity there is to create efficiencies and savings. What you also might not realize is evaluating your current cloud setup and usage is the simplest and quickest way to accomplish your goals.

1) Review Reserved Instances, Saving Plans, And Sustained Use Discounts

2) Spot Instances

3) Retiring Old Versions Of Instances

4) Autoscaling Of Apps And Infrastructure

5) Serverless Services

6) Keep An Eye On Data Transfer Charges

7) Look Out For Newer Services And Optimizations

8) Finish Your Pocs Faster

9) Continuously Analyzing Usage Patterns

10) Work With Cloud Providers

1 Review Reserved Instances, Saving Plans, And Sustained Use Discounts

Cloud services are popular for their on-demand per-second billing (pay per use). The on-demand pricing works great for trying out new cloud services and proof of concepts. For any longer usage, explore discount plans from cloud providers. AWS, Azure, and Google Cloud provide “Reserved Instances” pricing for many of its services on a 1-3-year contract that offer up to 75% discounts compared to on-demand pricing. AWS has also launched a new pricing model called “Savings Plan” that provides even better savings on the overall compute usage (EC2, Fargate, and Lambda).

On the other hand, Google Compute Engine service automatically applies a discount up to 30% based on the sustained usage without any contract agreement. Adopting these discounts is one of the most straight-forward ways to reducing the costs and requires no technical involvement. Everyone must use it, no exceptions.

An Egen customer was able to reduce costs by $2M+ annually by doing this!

2 Spot Instances

Spot instances are the virtual machines available to customers at 90% of the discounted cost compared to on-demand rates for a shorter lease duration (3+ hours - 1 day).

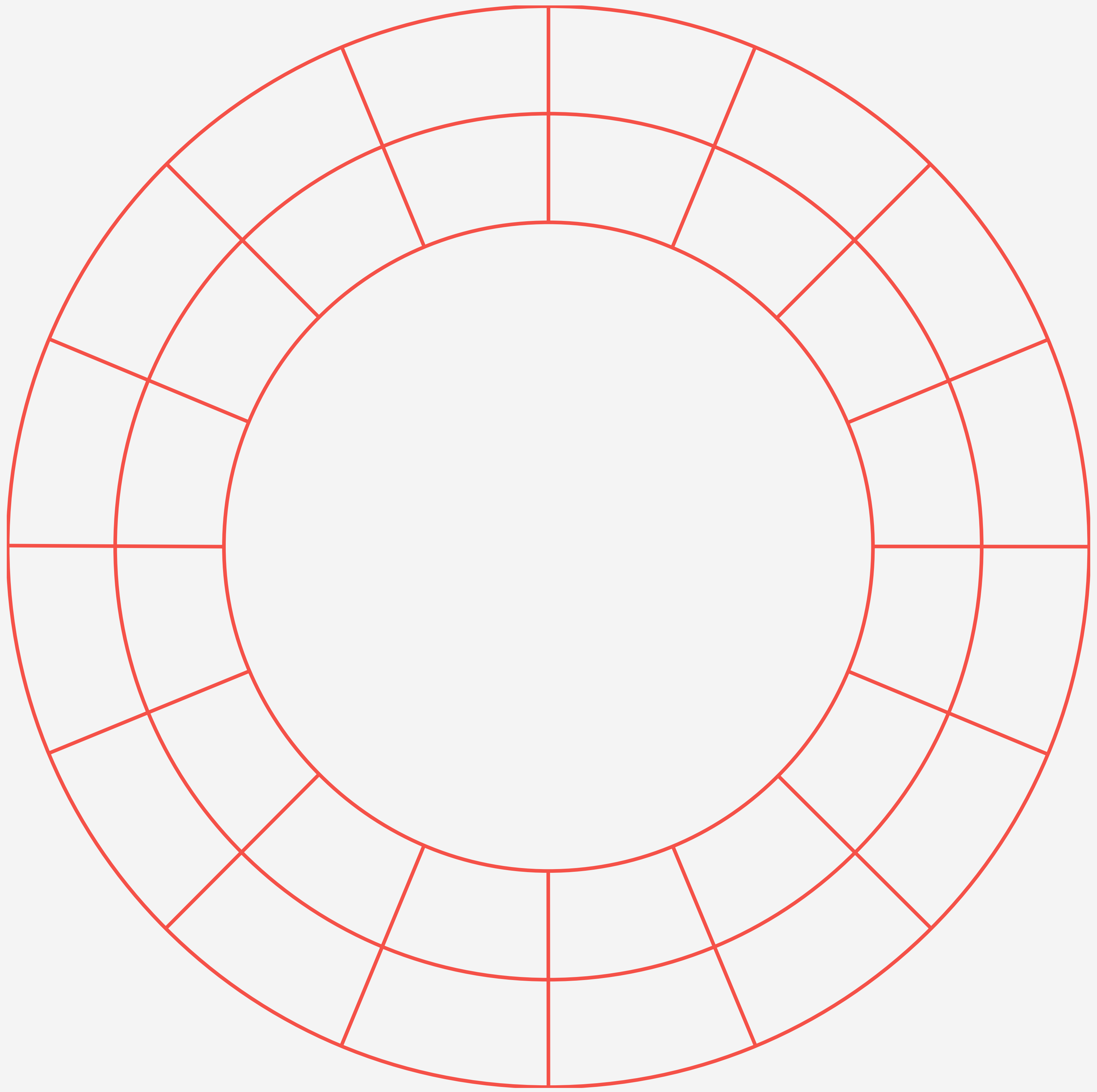
This instance type is perfectly suitable for batch jobs, developer tools, and stateless workloads that can be moved anywhere else within 30 seconds of the termination notice.

Utilizing these instances though requires an infrastructure setup where the workload can be automatically moved somewhere else.

3 Retiring Old Versions Of Instances

The cloud providers encourage customers to move on to the newer versions of virtual machines and other customer-managed resources (i.e. Databases, Data Warehouses etc.). Cloud providers are constantly upgrading their hardware underneath and working on reducing their own operating costs. When they are successful, the cloud providers usually transfer that cost savings to the customers. AWS is a great example here. Every year, they launch new generations of EC2 instances and reduce the cost compared to the previous generation. Now it's up to the customers to move their workload to a new generation of instances.

So, what's so difficult about that? Well, unless customers have put efforts in provisioning infrastructure using the principle of Infrastructure as Code ("IaC") from the start, moving the platform to new generation of VMs is a highly time-consuming task for the DevOps team. Their time spent here will outweigh the cost benefit of moving to the new generation of instances. In short, the customer is going to pay for their old-school infrastructure provisioning. If provisioned using the IaC principle, one should be able to create brand new environments or replace existing ones in less than hour without bringing down the platform itself. This is another example where having expertise is required to navigate the complexities.



4 Autoscaling Of Apps And Infrastructure

One core benefit of adopting the cloud is the capability of elastic scaling (both up and down) of compute services based on the system load and utilization. It would be wildly foolish to pay for over-provisioned resources 24x7 if the majority of the load on the system is only 2-3 days per week or few hours per day. Configuring “autoscaling” is not trivial for apps deployed on the cloud providers and requires expertise.

With proper autoscaling configuration, systems can be scaled up and down to achieve a perfect balance between high-availability and cost efficiencies. Another huge cost benefit can be achieved by shutting down non-production environments during the night and/or weekends. Having expertise in auto scalable infrastructure provisioning is a need-to-have in 2020.

5 Serverless Services

Humans are bad at estimating many things, and this axiom certainly applies when we are tasked to estimate cloud resources (CPU, memory, and storage) in advance. Most people would want to over-provision cloud resources to account for peak demand. Depending on the product though, peak demand may never come. Remember that everyone is not as popular as Uber, Airbnb, or Netflix. Keep in mind that the company has to pay for the over-provisioning of course, and that's where the new model of "Serverless Computing" comes into picture. Set up autoscaling properly for the apps and then off load all the server management work and risks to the cloud providers (they will always do a better job at managing servers than your DevOps team because it's their own infrastructure). Serverless services are also opening up new consumption models where the usage is easier to estimate than calculating CPU/Memory requirements.

For example, AWS Lambda (Functions as a Service) charges based on the number of invocations of the functions, which ties to the number of requests made to the API. That in-turn can approximately translate to number of users accessing your apps. Another example can be in the CI/CD pipelines. AWS CodeBuild and Azure DevOps charge based on number of build minutes per month. That translates to how many developers you have, multiplied by number of builds they are shipping every day (and month). No need to provision something for 24x7 when it's not going to be used that often. Developers don't work in the nights or the weekends (most of the time anyway, right?), so why pay extra for running dev tools during those times?

Now there is a caveat that is mostly applicable to larger workloads (typical for enterprises, high growth startups etc.). The serverless pricing is tricky for high consumption platforms and there is a limit for when serverless will be cheaper vs. managing servers on your own.

For example, when running containers, AWS Fargate (a serverless container service) will be cheaper until you can utilize 70% of the resources on an equivalent EC2 virtual machine. If you usually cross that threshold, running your infrastructure on your own will be more cost-effective. But this caveat is usually not applicable to all the customers. Only a select few.

6 Keep an Eye on Data Transfer Charges

Many cloud providers (AWS being the most notorious) charge for moving data between regions, zones within a region, and out of the cloud provider.

Plan the infrastructure networking following the least expensive routes so that a suitable balance among high availability, security, disaster recovery, and cost can be achieved. Maximize traffic that stays within a zone or at least a region.

7 Look Out for Newer Services and Optimizations

Constantly keep looking for new optimizations on the infrastructure side. Cloud providers are churning out new services and consumption patterns every few months. Pay close attention and see if anything released recently fits your needs. For example: Egen has been running a large Elasticsearch environment for a customer for 5 years now (~$75,000 per month). Elasticsearch is a costly service to maintain on your own. Two months ago, AWS announced the general availability of its Elasticsearch Ultrawarm capability. This new feature can provide 70-90% of the cost savings for the read-only data stored within Elasticsearch.

Additional savings can be achieved by the automatic archival of the older data using the Ultrawarm configuration. Surprise, surprise, the existing environment has 50% of its current Elasticsearch storage marked as read-only data (logs, output from machine learning models, and other historical indices) - huge savings.

We did a POC last month and have proposed this service to the customer now because who doesn't love saving money?? They are onboard and now we are moving the hosted Elasticsearch there with Ultrawarm config enabled. Combine this savings example with other ways mentioned here, and you are paying only a fraction compared to what they have been paying for the last 5 years.

8 Finish your POCs Faster

Don’t just keep hanging there with POCs lurking around for months. That money is never coming back.

More importantly, provision your POC infrastructure using an IaC tool like Terraform, AWS CloudFormation, Azure ARM or similar so that you can tear down all resources with a single step and bring back on whenever its needed. Work with the experts and find the best fit for your product as soon as possible.

9 Continuously Analyzing Usage Patterns

Make sure to look at your monthly billing statement. Billing dashboards are the single source of truth of your Cloud usage patterns. Analyze those and then walk backwards to see what services are over-utilized and for how long.

While Azure and Google Cloud have two of the best cost management tools for the cloud spend, keep in mind that AWS lags tremendously in this capacity. Don’t forget to set up cost alerts so that you don’t have unexpected expenditures each month.

10 Work with Cloud Providers

If you are a very high-usage cloud customer, work with the cloud provider's sales team and get your discount. Another thing that startups can utilize are the free credits available from the cloud providers for the startups.

For example, AWS Activate (https://aws.amazon.com/activate/) provides $100k credits to startups that qualify.

Bonus Tip: Pack Smart and Travel Light

This might be a bit controversial. Unless there is a strong case for it, run your product/platform by using open-source frameworks/tooling and not tying deep into the cloud providers' own proprietary application and data services. In short, avoid vendor lock-in. You can pretty much choose a multi-cloud approach by utilizing lowest common denominator services (services that are available on all the cloud providers). Examples include Kubernetes, Docker, databases like PostgreSQL, object/file storage, and networking.

This frees you from being locked-in and then you can jump the ship whenever a cheaper boat is available. If you are setup for this, you can evaluate the costs every day, and shift the workload automatically. For example, Zoom moved their workload to the Oracle Cloud to avoid high outbound data transfer fees from other cloud providers. Oracle gave Zoom a huge discount compared to the other 3 providers. Zoom mostly runs their platform on common compute and storage services (and hence has a portable workload), so moving to the cheapest bidder was an easy decision. Next month if AWS provide that discount, Zoom may very well move back.

egen

Schedule a free 30-minute consultation with Egen to see how you can uncover savings with your existing cloud structure.

https://egen.solutions/contact