QBurst

Cloud Cost Optimization

Best practices and learnings to maximize returns from your cloud investments

Cloud cost optimization is all about minimizing cloud spend by incorporating best practices such as identifying and downsizing underutilized resources, eliminating waste, using discounted instances, and designing workloads for scalability.

Pricing for public cloud services depend on resources ordered, and remain unchanged when not in use. The majority of cloud spend goes towards compute and storage resources, therefore it is key to optimize cost in order to maximize return on investment (RoI) and reduce total cost of ownership (TCO).

Compute

Track utilization

- Low CPU usage (idle instances).
- Identify unused/zombie instances and remove them. Ensure adequate controls are in place to manage sprawl.

Track over-provisioning

- Monitor larger than required instances and consolidate low CPU utilization instances onto fewer instances.
- Reduce network load on cloud servers by offloading web traffic to a content delivery network (CDN).
- Newer instance generation ensure higher performance at lower costs — switch where possible.

Right-size instances

- Instance types are optimized based on requirements such as memory, database, computing, graphics, storage capacity, and throughput.
- Experiment and pick the one that’s right for the workload.

Use auto-scaling to your advantage

- Take the smallest instances and auto-scale as needed.
- Find average and peak resource consumption for setting up auto scaling.

Reserved Instance (RI)

- Investment in RI could save money — especially for stable unchanging workloads.
- Ensure RI discounts are always applied and not wasted.
- Do not purchase RIs without visibility.

Spot instances/preemptible VMs

- Rearchitect your batch/cron jobs to use spot instances, however, these may be terminated anytime.
- Use spot instances with auto-scaling groups.
- Spot instances are not suitable for perpetually running, stateful applications, and databases.

Make use of scheduling

- Leverage automation to schedule instances to start and stop.
- Shut them down when resources are not being used.
- Use heat maps to check usage.
- For short term projects, manage VM/instance lifecycle by assigning an expiration or decommissioning date to each virtual machine that is created (unless it runs a critical workload). For example, development and test environs.

Technical recommendations

- Software stack: Instances running Linux are priced lower than RHEL and Windows Servers, therefore it makes sense to choose Linux. .NET applications can be migrated to .NET core to enable them to run on Linux systems.

- Optimize license costs: If using MS SQL, choose the right version — MS SQL Web/ Standard/Enterprise.

- Consider migrating to Postgres from Oracle and MS SQL. Tools such as AWS Schema Conversion can be used for this.

- In Bring Your Own License (BYOL) scenario, if the license is priced on a per CPU basis, consider disabling Hyper-threading to save on license costs.

- If you have already purchased Windows licenses, you might want to consider taking dedicated hosts to take advantage of existing licenses.

- Machine learning workloads: Consider using Elastic Inference instead of P type.

- If you require GPU power, an EC2 instance with Elastic Graphics might be better than G type instance.

- Lambda: Being serverless, Lambda can bring in operational efficiencies by saving administrative costs. However, it is more cost effective for time-limited, low-memory, stateless workloads. Failed Lambda executions cost money and retries are charged.

- Containers: If done right, using containers can improve utilization. Use lightweight OS images in containers.

- Containers on EC2 vs Fargate: AWS costs for Fargate are higher that running containers on EC2, but you save on administrative costs.

Storage

While putting in place a storage lifecycle policy, there are a few considerations that need to be taken into account.

Retention period: Determining the value of an object and its shelf life goes a long way in ensuring costs are in check. Objects can be tagged and rules created to delete storage classes after the expiration date or when the minimum threshold (based on compliance requirements) is reached.

Access patterns: Retrieval costs on long-term storage classes can be expensive. Storage class decisions should be made after evaluating access frequency/patterns and object value. Objects in long-term storage that have the potential to be frequently visited over time can copied and made available in a regional storage class to reduce high access charges.

Retrieval time and performance considerations: Consider where the object is going to be accessed from before making a decision on the storage class to be used. Objects meant for global consumption/multiple regions should be made available in the same region as the calling and retrieving resource to ensure higher availability, improve performance, and reduce costs.

Configure rules for data deletion/migration: Configure rules to delete objects that are irrelevant/obsolete. Delete incomplete multipart uploads and migrate data that shows signs of infrequent access. Identify and remove unused or unattached resources and dated snapshots that are no longer required.

Compress/optimize object size: Compress object size before storage. This enables files to be stored using less space and transferred faster, resulting in reduced storage and transfer costs.

Technical recommendations

- Object storage (S3): Choose S3 standard vs S3 Infrequent Access (IA) vs S3 One Zone IA.

- Use Intelligent Tier to move data between various S3 tiers based on usage.

- Use low-cost storage classes such as Amazon Glacier to backup data that is used infrequently, but required in the long-term to meet compliance and regulatory requirements. Use bulk retrieval with Glacier.
- Significantly reduce transfer and compute costs by using tools such as S3 select to selectively retrieve objects.
- S3 lifecycle management: Move between tiers, expire old versions.
- Gateway VPC endpoint ensures traffic remains internal.
- Automate time-consuming storage administration tasks with Amazon EFS. Pay for the space that the files and directories use — no minimum fee or setup cost.

Network

- Minimize cost of internal data transfers: Reduce data egress fees by creating a framework that limits data transfers. Avoid constant data transfers from the cloud to on-premises applications — consider migrating such applications to the cloud.
- Use private IPs or IPV6 instead of public IPs.
- Idle load balancers: Unused Elastic Load Balancers (ELBs) accrue charges. ELBs that are not attached to an instance or the ones that have a low request count should be removed.

Applications

Going with a single cloud or multi-cloud strategy have their own pros and cons. The benefits of each are listed below.

Single Cloud	Multi-cloud
• Volume discounts as a result of limiting cloud investments to a single provider. Cloud service providers offer discounted prices to customers that purchase products or services in higher volumes. • Eliminates the administrative hassles of switching between platforms, costs associated with network traffic between clouds, and training on multiple platforms. • Centralize AWS account - efficient use of RIs.	• Compare cloud costs as network and data transfer charges vary from vendor to vendor. Achieve a balance between cost and performance. • Dedicated network connection services offered by the respective cloud service provider (AWS Direct Connect) can help to keep data transfer charges in check.

Governance

Cloud Governance solutions enable enterprises to control costs with a high level of visibility into risk, compliance, and security. Implementation of a cloud governance framework improves decision-making with standardized policies for departments and roles. It facilitates orchestration of resources and services across technologies and business functions.

FinOps

FinOps provides a framework to manage budgets and costs towards cloud services. Enterprises are now adopting FinOps to break down silos between cross-functional teams and to gain a better understanding of variable spend on the cloud. FinOps enables enterprises to:

- Allocate cloud costs to the appropriate cost centers or teams.
- Leverage tags for chargeback and showback to allocate costs to departments or business units.

The approach helps enterprises to gain a broader perspective on consumption-based spending. The important thing to remember is that cost optimization is an ongoing process and it takes continuous and concerted efforts from the team to achieve this goal.

Cloud Cost Optimization Assessment

Cloud cost optimization is a continuous process. Our Cloud Cost Optimization Assessment service involves a review of your cloud infrastructure followed by a report and recommendations to eliminate waste and minimize costs. The report covers information on aspects such as instances, resource utilization, reserved instances, and service changes.

The report is created by a team of cloud consultants skilled in cloud cost optimization tools and methodologies. This service is available for companies using AWS, Microsoft Azure, and Google Cloud Service.