The cloud economics playbook for financial services

A 3-6-9-3 framework to demystify cloud cost optimization

/thoughtworks

Design. Engineering. AI.

Contents

Getting the cloud under control

Cloud services are rapidly becoming indispensable for banks and other financial services providers. Almost all - 98% - of financial services (FS) firms use cloud computing in some form, according to the Cloud Security Alliance. Most of these firms see significant advantages over traditional on-premises IT infrastructure. In a recent poll by LSEG (London Stock Exchange Group), 61% of senior FS executives report that cloud adoption contributed to a reduction in their IT cost of ownership (TCO). Other major benefits include:

- **Scalability and agility:** Cloud services provide on-demand resources that can be easily scaled up or down based on business needs.
- **Faster experimentation and innovation:** Cloud services provide access to cutting-edge technologies and tools that can help enterprises accelerate experimentation, leading to innovation and more efficient development of new products and services.
- **Reduced operational overhead:** By shifting IT infrastructure management to the cloud, FS firms can free up valuable IT resources to focus on core business activities and strategic initiatives.

All these capabilities have become critical in an environment where banks and other FS providers are facing rising customer demands for seamless, real-time services, and striving to compete – and/or cooperate – with a raft of new, digital-first market entrants. Studies show that banks see fintech and big tech firms displacing other major banks as their biggest competitive threat over the next 6-10 years – but they also view the ecosystems and embedded services made possible by partnerships with these firms as their greatest non-core revenue opportunity.

To capture this momentum, cloud-native digital banking platforms will be essential. Yet too often banks' cloud-based transformations remain tentative. Less than a third of FS firms have a majority of business-critical workloads in the cloud, for example.

A variety of factors prevent banks from embracing the cloud fully. Policies like the General Data Protection Regulation (GDPR) in Europe and the Gramm-Leach-Bliley Act (GLBA) in the US affect what data can be stored where. Many financial institutions are contending with tangled legacy systems that have been stitched together over decades, based on programming languages that don't translate easily to today's cloud services.

One particular pain point is becoming more and more apparent: cost. While cloud migration is often painted as an opportunity to consolidate and reduce IT spending, this is not always the case in practice. Banks often underestimate the costs of cloud migrations, and a significant majority of firms end up overshooting their cloud spending targets. Among the main reasons:

- **Underutilized and wasted resources:** The on-demand nature of cloud can be a double-edged sword. While it allows for easy scaling, it's also easy to overprovision resources that remain idle, or leave resources unused.
- **Failing to recognize the indirect cost:** Cloud services offer a vast array of features and options, making it easy to unknowingly incur unexpected charges such as DataTransfer, Observability etc.
- **Shadow IT:** Experimentation and innovation are inherent aspects of a thriving cloud infrastructure. However, when developers spin up cloud resources outside of established guidelines to explore new services, it can lead to a lack of cost visibility and control — a practice often referred to as Shadow IT. This also poses security and compliance risks for financial institutions without sufficient controls and guidelines in place.

- **Unpredictable budgets:** Uncontrolled cloud spending can wreak havoc on financial planning as billing post facto. Without proactive cost management, enterprises face surprise spikes in bills and struggle to forecast future expenses accurately.

AI adoption will only increase cloud usage and the amount banks need to invest. No less than 91% of global financial services firms are already using or planning to use cloud services for AI initiatives over the next 12 months, according to LSEG's survey. In this context, **cloud cost management** is becoming the cornerstone of a successful and sustainable cloud strategy for FS firms. With effective cloud cost controls, FS firms can leverage the benefits of cloud computing without getting bogged down by financial mismanagement.

In this Playbook, we set out a '3-6-9-3 strategy' of optimal cloud economics that ensures you can manage cloud costs and operations effectively, positioning your organization to take full advantage of the innovation and partnership opportunities arising in an increasingly fluid and technology-enabled FS landscape.

Understand	Avoid
3	6
types of services	beginners mistakes
Apply	**Uphold**
9	3
optimization strategies	pillars of governance

3

Three types of cloud services

Cloud services can be broadly categorized into three types, offering flexibility in deploying workloads. The best choice for your organization depends on many parameters, from application specific needs, in-house skills in managing operations, usage patterns and cost structure.

Serverless services

Serverless services eliminate operational overhead, including provisioning, configuration, maintenance and capacity planning. These services offer fast automatic scaling with a pay-per-use pricing model. However, they can become costly with higher usage.

Examples: SaaS offerings from cloud providers such as AWS Lambda, EMR Serverless, DynamoDB, SQS and SNS, AWS Fargate, and AWS ECS.

Pro tip: Use a serverless service such as AWS Lambda for short-duration tasks that run infrequently, such as automating daily policy checks.

Managed services

Managed services enable cloud vendors to provide their expertise and automation to provision and manage the operational aspects of the service or application. Managed services provide ease of operations and cloud portability, and are generally cost effective. However, only a few popular open source frameworks are available as managed services.

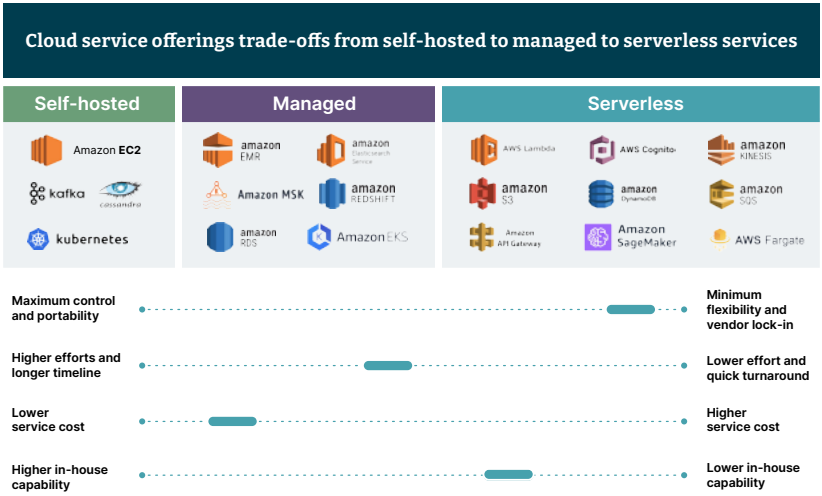
Examples: Managed Kubernetes (K8) clusters, like Amazon's Elastic Kubernetes Service (EKS); managed relational databases (RDS); managed Redis cache; managed OpenSearch; and managed Kafka, such as Amazon Managed Streaming for Apache Kafka (MSK).

Pro tip: Compared to traditional x86 options, many AWS-managed services that use Graviton (ARM-based) processors deliver better price-performance ratios and reduced environmental impact. One example is AWS MSK.

Self-hosted vanilla service

A self-managed service is one that runs on a pure vanilla virtual machine (e.g. MariaDB on EC2). This choice brings maximum flexibility with the tech stack, but demands higher effort and longer timelines, as it leaves the organization responsible for patches, fixes, backups, availability, software upgrades, dependencies, network infrastructure, security and so on. Cost-wise, self-hosted is cheapest, but requires significant operational overheads.

Pro tip: Self-hosting is best suited for organizations with strong in-house cloud infrastructure management expertise.



Source: Thoughtworks

Which service should my organization use?

Craft your cloud deployment architecture by carefully selecting services that match your workload requirements while remaining cost-effective. Consider a hybrid cloud approach for optimal cost and flexibility.

- **For rapid prototyping, short-lived or low-usage workloads, or variable or unpredictable loads:** Opt for serverless services to minimize effort and expedite development. While serverless options can be costlier, the benefits outweigh the expense in these scenarios. However, be mindful of potential vendor lock-in.
- **For high-utilization workloads:** Choose cost-effective options like self-managed or managed services. These offer more flexibiilty and avoid vendor lock-in, but require in-house expertise for provisioning and maintenance.

Pro tip: Strive for an 80/20 cloud service mix to optimize flexibility and minimize vendor lock-in. For 80% of workloads, build application architecture with cloud-agnostic services. This promotes portability and reduces dependence on a single cloud provider. Allocate the remaining 20% of workloads to cloud-specific services for best value, ease and strategic use cases. This enables you to leverage the unique value and capabilities that cloud providers offer for specific use cases, while maintaining overall flexibility.

6

Six common pitfalls

The cloud offers immense flexibility and scalability, but can also lead to unexpected costs if not managed properly. Here are some common pitfalls that can lead to unexpected cloud bills, and failure to realize the anticipated returns of cloud adoption. By remaining mindful of these pitfalls, and implementing best practices (covered in the following section, 'Nine techniques of cost optimization'), you can significantly reduce cloud costs and maximize the value of your cloud investment.

Overprovisioning and unused capacity	**Failing to recognize the indirect cost**
Long running experiments	**Fear of commitment based savings**
Not upskilling the workforce	**Old ways of working**

1. Overprovisioning and unused capacity

In the context of cloud costs, overprovisioning and unused capacity both lead to inefficient spending. For those coming from an on-premises world, it can be tempting to allocate more resources (compute, storage, etc.) than you need or use. Yet from an efficiency perspective, this is like renting a mansion when you only need an apartment.

Common causes include:

- Over-provisioning for peak loads.
- Static resource allocation.
- Lack of rightsizing and forecasting.
- Ignoring utilization metrics.

2. Failing to recognize the indirect cost

The underlying complexity of cloud service charges often leads to budget overruns. In fact, research from EY suggests almost 60% of organizations exceed their annual cloud budgets.

Unlike on-premises infrastructure, where network usage costs are generally not a concern, cloud environments introduce additional expenses that are frequently underestimated. Specifically, many banks accustomed to on-premises infrastructure will never have encountered data transfer costs between Availability Zones (AZs), Virtual Private Clouds (VPCs) and regions – all of which can be substantial. Not factoring such costs into cloud budgets can result in unexpected charges and nasty surprises.

Common causes include:

- Lack of centralized visibility over costs, particularly in hybrid/multi-cloud environments, which research shows are used by over 80% of financial institutions.
- Extended support costs with managed services.
- Ignoring data transfer and networking costs.
- Not estimating observability costs, such as Cloudwatch for AWS.
- Additional expenses to make necessary provisions for security and compliance, especially as regulators like the UK's Financial Conduct Authority tighten scrutiny of and the rules around banks' cloud usage.

3. Long-running experiments

With virtually unlimited on-demand capacity, teams can fall into the trap of running experiments indefinitely, or forgetting to terminate resources after use — much like paying for that fancy gym membership that you never use.

Common causes include:

- Never-ending experiments, or forgetting to tear-down after experiments have concluded.
- Insufficient rigor in deciding which experiments get funded.

4. Fear of commitment-based savings

Businesses often worry about committing to a specific level of spend or instance type for a one- or three-year term, fearing they might underutilize resources if their needs change. Conversely, they might outgrow the reserved capacity, necessitating sudden additional spending.

Common causes include:

- Lack of forecasting due to uncertainty about future needs and poor capacity planning.
- Fear of vendor lock-in.

5. Not upskilling the workforce

A cloud-based environment requires different skills for managing and optimizing costs. Moving from manual on-premise processes to fully automated on-demand cloud infrastructure requires coding proficiency and a new mindset.

Common causes include:

- Lack of capability-building programs to upskill the workforce.
- Not understanding cloud complexity and pricing structures.
- Inefficient coding practices for Infrastructure as Code (IaC).

6. Continuing old ways of working

Migrating traditional IT practices to the cloud without adaptation often leads to inefficiencies, negating the cloud's primary advantage — its on-demand, elastic nature. Organizations that "lift and shift" old methods may end up over-provisioning resources for peak loads instead of leveraging auto-scaling, for example. Additionally, manual provisioning and patching processes often result in 'snowflake' servers (unique, difficult-to-reproduce configurations) rather than standardized, immutable infrastructure. Manual operations result in configuration drift and increased complexity and costs.

Common causes include:

- 'Lift and shift' migration strategies preventing the optimization of resources.
- Manual provisioning and patching processes introducing additional complexity.
- Ticketed centralized manual processes and lack of automation (IaC).

9

Nine cost optimization strategies

Drawing from our experience working with numerous large financial institutions, we've developed a tested set of cloud cost optimization strategies designed to maximize cloud efficiency and minimize expenses. At one major bank we worked with in India, for example, adopting these strategies resulted in the identification of over 120 cost savings opportunities, and total savings of US$7 million annually.

Commitments	**Rightsizing**	**Autoscaling**
ARM processors	**Storage**	**Managed services**
Networking and data transfer	**Private pricing and MAP**	**Cost observability**

Commitments that create significant cost savings

Compute (i.e. EC2) instances often dominate cloud bills. To address this, cloud providers offer commitment-based discounts like savings plans (SP) and reserved instances (RI). These can potentially cut compute costs by 30%, and even up to 50%. However, these savings apply only to compute resources - not other cloud services such as storage or managed services.

Follow this three-step process to purchase SPs and RIs:

1. **Analyze usage patterns.** Review your cloud resource usage over time to identify consistent, predictable workload patterns.

2. **Choose the right commitment type.**
    - Savings Plans. These are more flexible, as they apply to multiple instance types/families across the EC2 instance families including EKS nodes, Lambdas as well as EMR Clusters.
    - Reserved Instances. These are specific to an instance type/family, and availability zone. Best suited for managed services for databases such as RDS.

3. **Regularly review and adjust:** Use cloud provider tools or third-party solutions to analyze commitment usage and readjust commitments by repeating steps 1 and 2. Also, remember to track plan expiration dates.

Start conservative, such as committing 50% of your baseline usage for a one-year term with no upfront payment option, and gradually increase commitments from there. Also, purchase SPs and RIs at organizational billing accounts for flexibility during AWS account restructuring.

EC2 Instance	Memory GiB	vCPUs	On demand	Reserved cost (3 years, No upfront)	Savings
m5.large	8	2	$0.10	$0.04	56.44%
m5.xlarge	16	4	$0.20	$0.09	56.93%
m5.2xlarge	32	8	$0.40	$0.18	56.68%
m5.4xlarge	64	16	$0.81	$0.35	56.81%
m5.8xlarge	128	32	$1.62	$0.70	56.81%

Source: instances.vantage.sh
(AWS Pricing for Asia Pacific Mumbai Region as of June 2024 with Savings)

Pro tip: If you end up committing to more resources than you need, you can resell your RI commitment on RI marketplace. Read more here.

Best practice: Avoid making large, upfront purchases of SPs or RIs, as this can lead to overcommitment. Consider making multiple smaller purchases gradually.

Additional resources: AWS Savings Plans, GCP CUD, Azure Savings Plan, Savings Plan vs Reserved Instances.

Rightsizing with effective resource utilization

Rightsizing resources is a crucial technique in cloud cost optimization because it ensures you're paying for exactly the computing power you need. Many cloud bills are inflated by overprovisioned resources due to a lack of capacity planning, and continuing old ways of provisioning. Rightsizing eliminates this waste by identifying and scaling down instances with low utilization. Based on our experience, we've observed that rightsizing can often lead to cost reductions in the range of 30-70% of compute spend.

Near real-time monitoring of your cloud resources' utilization of CPU, memory, storage and network usage is essential for effective rightsizing. This monitoring provides valuable insights into usage patterns over time, including peak and off-peak periods. By analyzing this data, you can identify resources with consistently low utilization rates, allowing you to optimize their size and save cloud costs.

Pro tip: Key to rightsizing is selecting the right instance family along with instance size. For example, in AWS choose R series instances for memory-heavy workloads like databases and caches.

Use burstable T series and leverage Spot instances for experimental, non-critical and lower environment workloads.

Best practice: Remember rightsizing should be an ongoing process that requires continuous evaluation and optimization, similar to constantly fine-tuning performance.

Additional resources: Adidas Case Study, AWS Compute Optimizer, AWS Instance Types.

Autoscaling to adapt to daily demand fluctuations

The cloud's beauty lies in its scalability. With rightsizing, you can easily scale resources up during peak periods and down during downtime. This eliminates the need to pay for unused capacity.

Problem: Static provisioning for dynamic demand

Traditionally, cloud resources were provisioned with a fixed amount of compute power, such as CPU and memory. This static approach can lead to inefficiencies. Overprovisioning occurs when you're paying for unused resources during low-demand periods, a common issue given that most workloads experience fluctuations. On the other hand, underprovisioning can cause performance problems during peak demand, potentially resulting in lost revenue or a negative customer experience.

Solution: Dynamic scaling with horizontal autoscaling

To address these challenges, dynamic scaling through a feature called horizontal autoscaling offers a more flexible approach. This feature automatically adjusts the number of instances (virtual machines) by adding or removing them based on predefined metrics or schedules. This ensures that your application scales in real time to meet demand.

Two autoscaling strategies: Metrics-based and schedule-based

1. **Metrics-based scaling:** This is based on defined thresholds for metrics such as CPU utilization or API Request traffic. When a metric breaches the upper or lower threshold, the autoscaler automatically adds or removes resources (e.g. virtual machines, containers) to match demand. Metrics-based scaling is good for unpredictable workloads.

AWS EMR provides metrics-based managed scaling as part of managed services.

2. **Schedule-based scaling:** This is based on a defined schedule (e.g. weekdays vs. weekends, hourly variations, or events like a standard payday that can impact demand for banks' services) and the desired number of resources for each time period. The autoscaler automatically provisions or removes resources based on the schedule. Schedule-based scaling is good for predictable needs.

Autoscaling automates resource provisioning, freeing up your team to focus on other tasks. Dynamic scaling with horizontal autoscaling is a game-changer for cloud cost optimization. Rightsizing pod resource requests in Kubernetes is similar to rightsizing EC2 instances resources.

Pro tip: To achieve efficient dynamic scaling of nodes in an EKS cluster, use tools like Karpenter that help not only with rightsizing the number of nodes, but also by choosing the best instance size for workloads deployed.

Antipattern: Relying solely on CPU utilization can lead to suboptimal scaling decisions. Consider using additional metrics such as network traffic, memory usage or custom metrics to get a more accurate picture of your application's resource needs.

ARM processors for general purpose workloads

ARM processors deliver the best price performance for general purpose applications. The ARM processors in each cloud service provider are Graviton for AWS, Axion for GCP and Ampere Altra for Azure.

The pricing of ARM-based processors is significantly lower than for Intel processors; however, not all software is currently optimized for ARM architectures. Before migrating, ensure your workloads are compatible with ARM processors.

API name	Memory	vCPUs	On demand	Graviton savings
m7g.xlarge	16	4	$0.16	19.05%
m7i.xlarge	16	4	$0.20	
m7g.2xlarge	32	8	$0.33	19.05%
m7i.2xlarge	32	8	$0.40	

Source: instances.vantage.sh
(AWS Pricing for Asia Pacific Mumbai Region as of June 2024)

Pro tip: Leverage Graviton processors for AWS Managed services such as RDS, MSK, EMR Clusters for a direct price-to-performance benefit of ~20%.

Storage optimization

Cloud storage offers a seemingly endless pool of space, but neglecting storage optimization can lead to significant and unexpected costs. Without actively monitoring and analyzing storage usage, it can be difficult to identify areas for improvement.

Storage optimization techniques include: Migrating EBS volumes from gp2 to gp3 type; rightsizing your EBS-provisioned IOPS rate; cleaning up detached EBS volumes; configuring storage lifecycle management policies and implementing intelligent tiering; choosing the right S3 storage class, such as Glacier storage, for archival purposes; cleaning up excessive backup snapshots; and compressing and deduplicating data.

Pro tip: Having a clearly defined data policy and governance strategy can significantly reduce costs associated with cloud storage.

Consider gp3 volumes, which offer baseline IOPS with the ability to burst for additional performance when needed. The difficulty in predicting IOPS needs can lead to the common pitfall of allocating more IOPS than your workload requires.

Antipattern: Bear in mind that large datasets are particularly vulnerable to storage inefficiency, with duplicate copies often created for different teams or purposes.

Additional resources: Amazon EBS.

Managed services optimization

While managed services reduce operational effort and the need for in-house expertise, they can also introduce the challenges of vendor lock-in and higher costs if not utilized effectively.

As a sensible default, we recommend that you choose managed services for open-source components such as Kubernetes, Kafka, Postgresql or Mysql RDS, Redis, etc. Understanding the cost of AWS managed services is crucial due to the complexity of its pricing, which involves a pay-per-use model, variable factors and managed service overhead. To save costs, consider using RIs (reserved instances) and Graviton processors.

Pro tip: Avoid the unnecessary cost of extended support for managed services by upgrading to the latest version. For example, the cost of EKS extended support is significantly higher than standard support.

Antipattern: Keep in mind that sending observability data from CloudWatch to a custom self-hosted or SaaS observability stack can incur additional costs for processing metrics as well as egress charges.

Additional resources: AWS EKS Version and Support.

Networking and data transfer optimization to avoid hidden costs

When calculating the cost of AWS resources, networking components and data transfer costs are the most consistently underestimated.

Knowing the purpose, use and associated charges of networking components is key to avoiding unnecessary networking costs. Choosing the right Gateway service from options including AWS Transit Gateway, NAT Gateway or Internet Gateway can be difficult as these components have different hourly rates and data transfer charges.

Data transfer costs in the cloud can be complex. Many factors influence pricing, making it difficult to predict exact costs upfront. Cloud provider cost tools like AWS Cost Explorer can make it easier to monitor and analyze data transfer usage and costs.

Pro tip: Multiple availability zones equal higher availability, but may not really be required. Assess the trade-off between increased availability and additional costs when implementing multi-AZ or multi-region architecture. Leverage service mesh zone aware routing and cell-based architecture to reduce data transfer costs. However, please note there is no such cost in GCP.

Antipattern: Avoid using NAT Gateway for inter-VPC and on-premises network connectivity, as it is much costlier than other options such as Transit Gateway.

Taking advantage of Private Pricing (PP) and Migration Acceleration Programs (MAP)

Migration acceleration programs (MAP) provide a one-time migration benefit along with a suite of free tools, services and expertise to streamline the migration process. These programs are designed to facilitate the transition of applications from data centers to cloud infrastructure until they are stabilized and optimized.

Since MAP is complex and frequently updated, staying informed and following all guidelines when claiming benefits is crucial. Similar migration programs exist for GCP and Azure as well.

AWS also offers private pricing agreements (PPA) or the enterprise discount (EDP) to high-volume, long-term AWS users who commit to a certain spend over a specified period (usually one to three years). Private pricing offers discounts on your overall AWS bill over and above your SP and RI commitment-based savings. If you have high, predictable AWS usage and are comfortable with a spending commitment, PPA can offer significant cost savings.

Pro tip: Certified AWS Migration Competency Partners like Thoughtworks can help you with your migration journey. We have a proven track record and the expertise to deliver large-scale migration projects. Find out more here.

Antipattern: Sharing the higher side of estimates on projected future spending to get better private pricing.

Cost observability dashboards

Effectively managing cloud costs requires a proactive approach. Democratizing to real-time cost observability dashboards provides valuable insights into your cloud spending, empowering application teams to make informed decisions and optimize costs. This practice allows you to:

- Identify cost trends over time and spot any sudden spikes that require investigation.
- Gain insights into how efficiently resources are being utilized, to identify potential areas for rightsizing or scaling down.
- Generate alerts for anomalies in cost, usage and utilization patterns.

Pro tip: Select a cost and utilization dashboard tool that aligns with your specific needs and offers the functionalities required. For example, does the tool provide a detailed analysis of container workloads inside the EKS cluster? Does it offer utilization analysis for EMR clusters running MapReduce workloads? Does it offer recommendations for optimization?

Antipattern: Avoid investing in a tool without understanding if it supports dashboards or recommendations for the type of workload that is running.

Additional resources: FinOps Tools

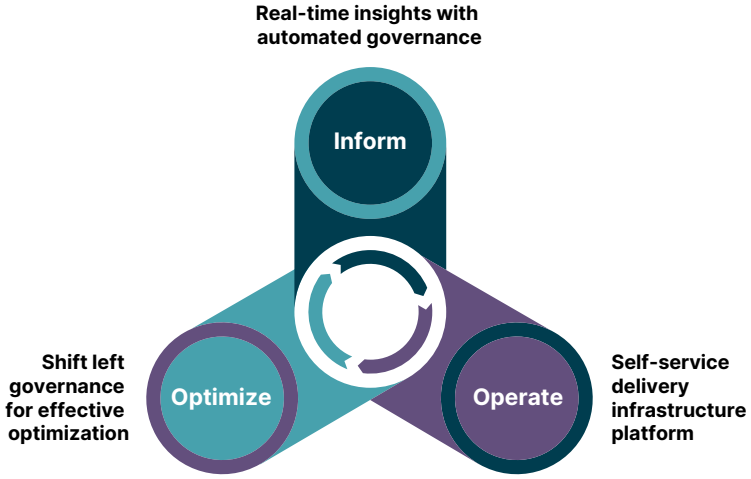
Disclaimer: This list of cost optimization techniques is deliberately not ranked; we recommend adopting them and establishing priorities based on your unique cloud service usage patterns.

3

Three pillars of effective cloud governance

Cloud governance is essential for organizations to ensure that their cloud infrastructure is used effectively, securely and efficiently, yet at many banks it remains a nascent practice. A survey by PwC found that in Europe, for example, less than a third of financial institutions feel they have adequate cloud controls in place.

By implementing effective cloud governance, banks can maximize the benefits of cloud computing while minimizing risks and costs. The following three pillars of cloud governance are aligned to the three phases of the FinOps framework.



Three pillars of cloud governance aligned to three phases of FinOps framework

Real-time insights with automated governance (FinOps: Inform phase)

Empower everyone to understand infrastructure and application health by providing dashboards with utilization, performance and spending data.

A few recommended dashboards detailing cost and utilization are:

1. **Spend breakdown**, with transparent cost visibility at the application, environment and cloud service levels, including chargebacks for shared services, SPs and RIs, support fees, and accounting for MAP Credits and Enterprise discounts.
2. **Utilization (usage) report** of CPU, memory and disks across all cloud services (EC2, EBS, EKS, RDS, EMR, etc.) and breakdown of resources by application, environment and microservices running within Kubernetes.
3. **Operational health dashboard**, showing performance-related data and security, sustainability and compliance statuses, as part of the overall health of an application.

Implement 'policy as code' to automate the enforcement of compliance, security standards and recommended practices. Implement fitness functions with alerts to ensure ongoing adherence and proactively identify and address potential risks. For example, write a fitness function for Amazon EKS that generates alerts when unsupported versions are running. To ensure swift action, clear guidelines should be established for non-compliance.

Pro tip: Implement fitness functions using serverless services like AWS Lambda and push data to relevant systems with actionable alerts.

Cloud-agnostic tools offer the flexibility to monitor both your on-premises infrastructure and cloud environments seamlessly.

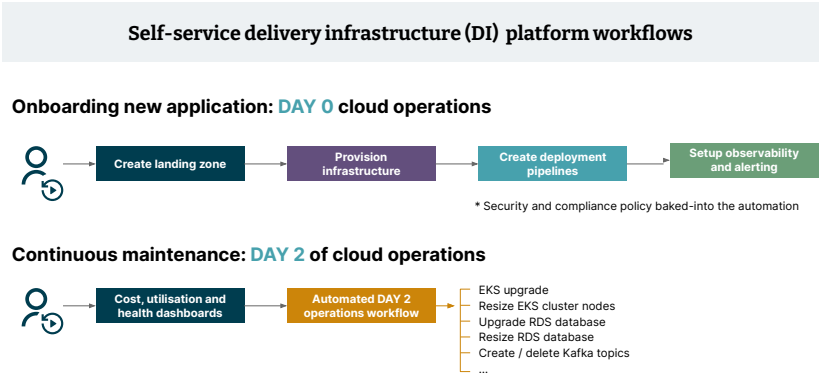
This enables comprehensive observability in a hybrid setup without bias or limitations.

Additional resources: FinOps Tools

Invest in a self-service delivery infrastructure (DI) platform (FinOps: Operate phase)

A self-service delivery infrastructure (DI) platform empowers developers and IT teams to provision, configure and manage their infrastructure environments (development, testing, production) on-demand, without relying on manual intervention from an infrastructure team.

By investing in a self-serving capability with best practices baked in, you can achieve faster deployments, improve consistency and reliability, and empower teams to be more efficient while operating at large enterprise scale. A self-service platform also helps with avoiding Shadow IT. Follow practices such as infrastructure as code and application templates while building a delivery infrastructure platform.



Pro tip: Leverage open source frameworks like Backstage.IO to build a Delivery Infrastructure Platform.

Shift left governance for effective optimization (FinOps: Optimize phase)

Running the first two pillars of cloud governance manually and centrally at enterprise scale would be highly challenging. Lessons from how cloud service providers operate suggest that these processes should be made self-service and delegated to application teams to streamline operations.

FinOps culture helps break down silos between finance, business and engineering teams. It also emphasizes aligning cloud spending with business goals; for example by measuring unit costs, such as cost per customer, feature or team. It's not just about reducing costs, but also optimizing resources to maximize the value delivered for the budget allocated.

Pro tip: By applying a rigorous tagging strategy to all cloud resources and accounts, you can enable precise cost allocation, in-depth spend analysis and actionable alerts to drive cost savings.

This approach ensures informed decision-making, efficient operations and ongoing optimization for a cost-effective and secure cloud environment.

Additional resources: FinOps Framework: Bringing accountability to cloud spend.

Metrics and KPIs for successful cloud cost management

Effective cloud cost management goes beyond just saving money. It's about optimizing your cloud resources to achieve the maximum value for your enterprise. Here are some examples of key metrics to measure success:

Metrics / KPI	Description
Cloud spend vs. budget (by application / project and environments)	Track your total cloud spend against your allocated budget. This helps identify areas of overspending and potential for cost reduction. Categorize applications / projects based on spend, e.g. S / M / L / XL. Automate daily, weekly and monthly spend tracking with overrun alerts. **Healthy state:** 100% of the cloud resources are tagged by application. <2% budget overrun, with quarterly budget revisions
Spend by services by application and environments	Analyze the cost breakdown of individual services by environments running in the cloud. This helps pinpoint areas where resources might be underutilized or highly expensive and allows you to look for alternatives. Monitor spend ratio from Prod vs NonProd environments. **Healthy state:** More than 80% of workloads are running with committed SP and RI. (Utilization of SPs and RIs should be 100%).

Metrics / KPI	Description
Resource utilization (CPU, memory and disk)	Monitor the average utilization of CPU, memory and storage resources across your cloud instances. Low utilization indicates potential overprovisioning and wasted spending. Cloud instances also include EKS Nodes, EMR Cluster Nodes and Containers like Pods running inside the K8s cluster. **Healthy state:** Ideally, EC2 instances should run above 70% utilization in cloud infrastructure, leveraging autoscaling to cater for peak loads and festive seasons.
ARM processor adoption percentage	ARM processors such as Graviton help reduce costs as well as carbon footprint, potentially contributing to sustainability goals. By default, use Graviton for MSK and RDS services. **Healthy state:** Having more than 25% of workloads running on Graviton with targets of a 2% increase per quarter.
Observability coverage	Gather observability data from all relevant sources, including applications, infrastructure components and network traffic. Look at all three aspects of observability - metrics, logs and traces. Also monitor the cost of observability infrastructure as compared to the cost of application hosting. **Healthy state:** Ensure more than 95% of resources of applications have cost observability setup. And in the best cases, the cost of observability should be in the 10-15% range, with the worst case not exceeding 25% of the application hosting cost.

3-6-9-3 Strategy of optimal cloud economics

"The cloud is transforming how businesses operate, requiring enterprises to adapt to a new model for consuming and managing infrastructure resources."

Sunit Parekh
VP, Digital Platforms Practice Lead,
Thoughtworks India

For optimal cloud economics, it's essential to understand the three types of cloud services and their trade-offs, avoid the six common mistakes that inflate cloud spend, and implement nine techniques for cost optimization while adhering to the three pillars of effective cloud governance.

The cloud landscape is constantly shifting, with new services and best practices emerging all the time. Staying ahead of the curve requires learning continuously about these advancements and their cost structures. By embracing migration as the default (even from one cloud service offering to another) and cloud economics best practices, financial institutions can leverage the numerous benefits of the cloud to gain agility and competitive edge. The '3-6-9-3' approach provides a foundation to explore innovative new products and partnerships, and seize the transformative potential of AI - without sacrificing security, compliance or financial discipline.

Author

Sunit Parekh

VP, Digital Platforms Practice, Thoughtworks India

sunitp@thoughtworks.com
@sunitparekh
sunitparekh.in

With over 20 years of experience, I'm a seasoned technology strategist passionate about helping clients achieve their digital goals. I specialize in guiding large enterprises through complex distributed projects, from global solutions to digital transformations. My expertise lies in crafting impactful technology strategies and implementing cutting-edge cloud-native solutions across ambitious projects.

Modern engineering advocate and cloud-native champion
I'm a firm believer in leveraging the power of cloud ecosystems and embracing cloud-native approaches to build modern, scalable infrastructure. I'm equally passionate about collaborating with clients who share my commitment to adopting modern engineering practices for achieving technical excellence.

Open source contributor
Beyond my client work, I actively contribute to the open-source community. I've built a valuable tool, Data Anonymization, that helps developers safely anonymize production data for testing purposes.

We are a global technology consultancy that delivers extraordinary impact by blending design, engineering and AI expertise.

For over 30 years, our culture of innovation and technological excellence has helped clients strengthen their enterprise systems, scale with agility and create seamless digital experiences.

We're dedicated to solving our clients' most critical challenges, combining AI and human ingenuity to turn their ambitious ideas into reality.

thoughtworks.com

/thoughtworks

Design. Engineering. AI.