hostitworks.com
24 Free Web Hosting Tools

$

CLOUD COSTS

Cloud Cost Optimization Playbook

2026 Edition

AWS · Google Cloud · Microsoft Azure

Cut your cloud bill by 40–70% with rightsizing, reserved instances, spot/preemptible instances, storage optimization & FinOps practices.

40-70%	3	50+	25+
Potential Cost Reduction	Cloud Providers	Cost-Saving Techniques	Pages of Playbook

Table of Contents

Chapter 8: FinOps Practices

- Tagging strategy
- Showback & chargeback
- Continuous optimization workflow

Introduction

The gap between what you pay and what you need to pay

Why Cloud Bills Spiral

Cloud providers make it effortless to spin up resources and unintentionally difficult to remove them. Engineers provision servers for experiments, forget to terminate them, and those servers run for months. Databases are sized for peak traffic that never materialises. Snapshots accumulate silently at pennies per GB until they represent hundreds of dollars per month. S3 buckets fill with logs no one reads. These are not edge cases — they are the median state of most cloud accounts that have been running for over 12 months.

Industry benchmarks consistently show that 30–35% of cloud spending is pure waste — resources that are idle, oversized, or redundant. Another 20–30% is suboptimally priced: on-demand instances running workloads that are predictable enough to qualify for reserved pricing at 40–72% discount. In aggregate, most organisations can cut their cloud bill by 40–70% without sacrificing any capability.

> **The FinOps Framework**
>
> FinOps (Financial Operations) is the practice of bringing financial accountability to cloud spending. Three phases: Inform (visibility — know what you spend and why), Optimize (act on waste — rightsizing, reserved instances, cleanup), Operate (continuous process — tagging, budgets, weekly reviews). FinOps is not a one-time project; it is an ongoing operational discipline.

The Optimization Hierarchy

Work through these categories in order — highest impact first:

Priority	Category	Typical Savings	Effort
1	Terminate idle resources	10–20% of bill	Low — find and delete
2	Rightsize oversized instances	15–30% of bill	Medium — monitor then resize
3	Reserved / committed pricing	20–40% of bill	Low — commitment purchase
4	Spot / preemptible instances	60–90% of compute	Medium — architecture change
5	Storage lifecycle policies	10–30% of storage	Low — set and forget
6	Egress & network optimization	5–20% of bill	Medium — CDN, architecture
7	Serverless where appropriate	Varies	High — refactoring required

Chapter 1: Compute Rightsizing

The fastest path to significant savings

Finding Oversized Instances

Rightsizing is identifying instances where actual resource utilisation is significantly below the provisioned capacity, and downsizing to a smaller instance type. The most common pattern: an engineer provisions an m5.xlarge "to be safe," the application runs at 8% CPU and 20% memory utilisation, and the server runs that way for two years. Dropping to an m5.large saves 50% of that instance cost immediately.

Utilisation Thresholds for Rightsizing

Rightsize candidates: avg CPU < 20% AND avg memory < 40% over 14+ days Rightsize urgently: avg CPU < 5% AND avg memory < 20% — likely idle Consider scale-down: avg CPU 20-40% — may benefit from smaller instance Leave alone: avg CPU > 40% or memory > 60% — appropriately sized

ARM-Based Instances — Up to 40% Cheaper

Instance Family	Provider	vs x86 Equivalent	Best For
Graviton3 (m7g, c7g, r7g)	AWS	~20% cheaper + 20% better perf	Most web workloads, APIs, databases
Graviton4 (m8g, c8g)	AWS	~30% cheaper than x86	Latest generation — best value
Ampere Altra (T2A)	GCP	~20% cheaper	Web servers, containerised apps
Ampere Altra (Dpsv5)	Azure	~20% cheaper	General purpose workloads

Most PHP, Python, Node.js, and Java applications run on ARM with zero code changes. Test your workload on ARM in staging — if it passes, the migration is a single instance type change. Combined with reserved pricing, ARM can reduce compute costs by 50%+.

30-50%

Switch to ARM + Reserved

Migrate to Graviton3/4 (AWS), Ampere (GCP/Azure) AND purchase 1-year reserved/committed savings simultaneously. Stack both discounts for the maximum combined reduction.

Auto Scaling — Pay Only for What You Use

If your traffic has predictable peaks (business hours, weekday vs weekend), running at peak capacity 24/7 wastes 50–70% of compute. Auto Scaling Groups (AWS), Managed Instance Groups (GCP), and VM Scale Sets (Azure) scale instance count dynamically with demand, so you pay for peak only during peak.

- Set scale-out threshold at 60–70% CPU — not 80%+ which causes latency spikes before scaling
- Set scale-in threshold at 20–30% CPU with a 10–15 minute cooldown to avoid thrashing
- Use scheduled scaling for predictable patterns: scale up at 8AM, down at 8PM
- Set minimum capacity to handle baseline traffic without scaling delay
- Use mixed instance policies: On-Demand for minimum, Spot for burst capacity

Chapter 2: Pricing Models

The same workload at 3 very different prices

On-Demand vs Reserved vs Spot — AWS

Pricing Model	Discount vs On-Demand	Commitment	Best For
On-Demand	None — full price	None	Unpredictable, short-lived workloads
1-Year Reserved (No Upfront)	~35%	1 year	Stable production workloads
1-Year Reserved (All Upfront)	~40%	1 year upfront	Maximum predictable savings
3-Year Reserved (All Upfront)	~60%	3 year upfront	Long-term stable infrastructure
Savings Plans (Compute)	~40-66%	1 or 3 year	Flexible across instance families
Spot Instances	60-90%	None (interruptible)	Batch, dev/test, fault-tolerant apps

40-66% savings

AWS Compute Savings Plans

More flexible than Reserved Instances — commitment is to $/hour of compute, not a specific instance type. Automatically applies to EC2, Lambda, and Fargate. Purchase in AWS Cost Explorer → Savings Plans → Purchase Savings Plans.

20-91% savings

AWS Spot Instances for Batch Workloads

Spot uses spare AWS capacity at 60-90% discount. Best for: CI/CD build agents, batch data processing, ML training, dev/test environments. Use Spot with Auto Scaling mixed instance policies for safe production use.

GCP Committed Use Discounts

CUD Type	Discount	Commitment	Applies To
Resource-based CUD	~37%	1 year	Specific vCPU/memory amounts in a region
Resource-based CUD	~55%	3 year	Same — deeper discount for longer term
Spend-based CUD	~17%	1 year	Cloud SQL, Cloud Run — commit to $ spend
Spend-based CUD	~22%	3 year	Same — deeper discount for 3 years
Sustained Use	Auto (up to 30%)	None	Automatically applied when instance runs >25% of month

Azure Hybrid Benefit + Reserved Instances

Azure has two major cost levers that many customers miss:

Up to 49% savings

Azure Hybrid Benefit

If your organisation has existing Windows Server or SQL Server licenses with Software Assurance, Azure Hybrid Benefit lets you reuse them on Azure VMs — up to 49% savings on Windows VMs and 55% on SQL Server. Activate in VM settings at zero additional cost.

Up to 72% savings

Azure Reserved VM Instances

1-year reserved: ~35% savings. 3-year reserved: up to 72% savings. Combine with Azure Hybrid Benefit for maximum stack discounts. Purchase from Azure Portal → Cost Management → Reservations.

Spot Equivalents Across Providers

Provider	Spot Product	Discount	Interruption Notice	Best Use
AWS	Spot Instances	60-90%	2 minutes	Batch, ML, CI/CD, stateless apps
GCP	Preemptible / Spot VMs	60-91%	30 seconds	Same as AWS Spot
Azure	Azure Spot VMs	60-90%	30 seconds	Dev/test, batch, fault-tolerant

Chapter 3: AWS Cost Optimization

EC2 · RDS · S3 · Lambda · Data Transfer · Cost Explorer

EC2 Cost Levers

- Use AWS Cost Explorer → Rightsizing Recommendations — shows specific instances to downsize with projected savings
- Enable Compute Optimizer (free service) for ML-driven rightsizing recommendations across EC2, EBS, Lambda
- Terminate instances stopped for over 7 days — stopped instances still charge for EBS volumes
- Delete unattached EBS volumes — they continue to charge even with no instance attached
- Release unassociated Elastic IPs — $0.005/hr (~$3.60/month) per unused Elastic IP
- Delete unused AMIs and their associated EBS snapshots — snapshots accumulate silently
- Use gp3 EBS instead of gp2 — same or better performance at 20% lower cost, requires no downtime

RDS Optimization

~60% savings

RDS Reserved Instances

RDS Reserved Instances provide 40-60% savings for 1-3 year commitments. Match the instance class to your actual utilisation — the same rightsizing rules apply: if avg CPU is under 20%, the instance is oversized.

- Stop non-production RDS instances on a schedule: dev/staging instances running 8h/day = 67% cost reduction
- Delete automated backups when terminating RDS instances — they persist and charge independently
- Use Aurora Serverless v2 for variable workloads — scales to zero during idle periods
- Consider moving read-heavy workloads to Aurora read replicas instead of scaling the primary
- Audit retention period: 7 days of automated backups is sufficient for most — every extra day costs storage

S3 Storage Classes

Storage Class	Cost (per GB/mo)	Retrieval Cost	Use For
S3 Standard	$0.023	None	Frequently accessed data
S3 Standard-IA	$0.0125	$0.01/GB	Infrequent access, retained long-term
S3 One Zone-IA	$0.01	$0.01/GB	Non-critical infrequent data, one AZ
S3 Glacier Instant	$0.004	$0.03/GB	Archives accessed once/quarter
S3 Glacier Flexible	$0.0036	$0.01/GB	Compliance archives, 3-5hr retrieval
S3 Glacier Deep Archive	$0.00099	$0.02/GB	Lowest cost, 12hr retrieval
S3 Intelligent-Tiering	Auto	$0.0025/1k obj	Variable access patterns — auto-tiers

S3 Lifecycle Policy Example

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket \
--lifecycle-configuration '{ "Rules": [{"ID": "auto-archive", "Status": "Enabled",
"Transitions": [ {"Days": 30, "StorageClass": "STANDARD_IA"}, {"Days": 90,
"StorageClass": "GLACIER_IR"}, {"Days": 365, "StorageClass": "DEEP_ARCHIVE"} ] }] }'
```

Data Transfer — The Hidden Cost

AWS charges $0.09/GB for data transferred OUT to the internet after the first 10GB/month. Cross-region transfer costs $0.02/GB. These charges are invisible until the bill arrives and can exceed compute costs for content-heavy applications.

- Route internet-bound traffic through CloudFront — CloudFront-to-internet costs $0.0085/GB (75% cheaper than direct EC2)
- Enable CloudFront for all S3-backed content — S3-to-CloudFront transfer is free within AWS
- Use VPC Endpoints for S3 and DynamoDB access from EC2 — eliminates NAT Gateway charges ($0.045/GB)
- Consolidate workloads to the same AWS region to eliminate cross-region transfer costs
- Use AWS Global Accelerator for global users instead of cross-region replication where possible

AWS Cost Explorer & Budgets

- Enable AWS Cost Explorer (free) — view spend by service, region, account, and tag
- Set AWS Budgets for monthly spend with email/SNS alerts at 80% and 100% of threshold
- Check Trusted Advisor weekly — identifies idle load balancers, underutilised EC2, unused EIPs
- Enable rightsizing recommendations in Cost Explorer — ML-powered, updated daily
- Tag all resources: Environment, Project, Owner, CostCenter — untagged spend is unmanageable
- Use AWS Organizations Cost Explorer for consolidated billing across multiple accounts

Chapter 4: GCP Cost Optimization

Compute Engine · Cloud SQL · Cloud Storage · BigQuery · GKE

Compute Engine

Up to 30% savings

Sustained Use Discounts (Automatic)

GCP automatically applies sustained use discounts when a VM runs for more than 25% of a month. No action required — the discount applies automatically and increases progressively: 25% of month = 10% off; 50% = 20% off; 100% = 30% off.

- Use GCP Recommender (free) — AI-powered rightsizing recommendations with projected savings
- Enable Committed Use Discounts for stable workloads: 37% (1yr) or 55% (3yr) savings
- Delete unattached persistent disks — check via: gcloud compute disks list --filter="users:(*)"
- Release unused static external IPs — $0.01/hr (~$7.20/month) per unused reserved IP
- Use Preemptible/Spot VMs for batch and fault-tolerant workloads (60-91% cheaper)
- Switch to E2 instances for dev/test — 31% cheaper than N1 for general purpose workloads
- Use custom machine types to precisely match CPU and memory needs without paying for waste

GKE Cost Optimization

- Enable GKE Autopilot — pay per Pod resource requests, not per node capacity
- Use Spot Pods (Autopilot) or Preemptible Node Pools (Standard) for batch workloads
- Set resource requests and limits accurately — oversized requests waste node capacity
- Enable Cluster Autoscaler with scale-to-zero node pools for variable workloads
- Use Vertical Pod Autoscaler (VPA) to right-size container resource requests automatically
- Check GKE cluster recommendations in Cloud Console — idle clusters are the most expensive waste

BigQuery Cost Control

BigQuery costs scale with data scanned per query — a single unoptimized query can scan terabytes and cost hundreds of dollars. Cost control is query efficiency.

- Always use column selection (SELECT col1, col2) — never SELECT * on large tables
- Use partition filters in WHERE clauses — partitioned tables scan only matching partitions
- Set query cost limits via maximum bytes billed per query: maximumBytesBilled parameter
- Use materialized views for repeated expensive queries — query the view, not the raw table

- Set BigQuery reservations (slots) for predictable workloads instead of on-demand pricing
- Preview estimated bytes before running large queries using --dry_run flag

Chapter 5: Azure Cost Optimization

Virtual Machines · Azure SQL · Blob Storage · Functions · AKS

Virtual Machines

Up to 55% savings

Azure Hybrid Benefit + Reserved

Stack Azure Hybrid Benefit (reuse existing Windows/SQL Server licenses) with 3-year Reserved Instances for maximum savings. A B2s VM drops from ~$38/mo to under $10/mo when both are applied simultaneously.

- Use Azure Advisor (free) — Cost tab shows specific VMs to resize, shut down, or reserve
- Enable auto-shutdown for dev/test VMs via Azure DevTest Labs or manual schedule
- Use B-series burstable VMs for workloads with low baseline CPU and occasional bursts
- Delete unmanaged disks attached to deallocated VMs — they charge even when VM is off
- Release unassociated public IPs — ~$3.65/month per unused Standard SKU IP
- Use Azure Spot VMs for batch workloads, CI/CD pipelines, and dev environments

Azure SQL & Storage

- Use Serverless tier for Azure SQL — auto-pauses after inactivity, billed per second of use
- Enable Blob lifecycle management policies to auto-tier data to Cool/Archive storage
- Use Reserved Capacity for Azure SQL and Blob Storage (1yr: 33% off, 3yr: 56% off)
- Enable Azure Dev/Test pricing for non-production subscriptions — saves up to 56% on VMs
- Audit Azure Blob soft delete retention — keeping deleted blobs costs storage at standard rate
- Use Azure Files Standard (HDD) instead of Premium (SSD) for infrequent access file shares

Chapter 6: Storage Optimization

Lifecycle policies, snapshot hygiene, object storage tiers

Object Storage Tiering — All Providers

Provider	Hot	Cool/IA	Cold Archive	Auto-Tier Option
AWS	$0.023/GB	$0.0125/GB (Standard-IA)	$0.00099/GB (Glacier Deep)	S3 Intelligent-Tiering
GCP	$0.020/GB	$0.010/GB (Nearline)	$0.004/GB (Archive)	No native auto-tier
Azure	$0.018/GB	$0.01/GB (Cool)	$0.00099/GB (Archive)	Lifecycle policies

70-95% savings

Move logs and backups to cold archive

Application logs and database backups are accessed infrequently after 30 days and almost never after 90 days. Automating tiering to cold storage reduces per-GB storage costs from $0.02 to under $0.001 — a 95% reduction.

Snapshot & Backup Hygiene

- Audit all snapshots across regions — unmanaged snapshots are the most common source of surprise storage costs
- Implement snapshot lifecycle policies: keep daily for 7 days, weekly for 30 days, monthly for 90 days
- Delete snapshots from terminated resources — cloud providers do not auto-delete them
- Use incremental snapshots where available (EBS snapshots, GCP persistent disk snapshots are incremental)
- Compress backup data before storage when using custom scripts — reduce S3/GCS/Blob storage by 60-80%
- Audit cross-region snapshot copies — replication to multiple regions multiplies storage costs

Chapter 7: Network & Egress

Egress is the most underestimated cloud cost

Data Transfer Type	AWS	GCP	Azure
Ingress (data IN)	Free	Free	Free
Egress to internet (first 10GB/mo)	Free	Free	Free (5GB)
Egress to internet (after free)	$0.09/GB	$0.08/GB	$0.087/GB
Same-region AZ to AZ	$0.01/GB	Free	Free
Cross-region transfer	$0.02/GB	$0.02-0.08/GB	$0.02/GB
Via CDN (CloudFront/Cloud CDN)	$0.0085/GB	$0.02/GB	$0.0075/GB

75-90% savings

Route traffic through CDN

Data transferred from EC2/Compute Engine/Azure VM to the internet costs $0.09/GB. The same data via CloudFront/Cloud CDN/Azure CDN costs $0.0085-0.02/GB — a 75-90% reduction. For high-traffic sites, CDN pays for itself many times over in egress savings alone.

- Place all resources in the same region to eliminate cross-region transfer costs entirely
- Use VPC/VNet peering instead of internet routing for inter-service communication
- Enable CloudFront/Cloud CDN for all S3/GCS/Blob-served content — origin-to-CDN is free within provider
- Compress API responses with gzip/brotli — reduces egress volume by 60-80% for text content
- Audit inter-AZ traffic patterns — within AWS, same-AZ communication is free; cross-AZ costs $0.01/GB
- Use PrivateLink / Private Service Connect to keep traffic within provider network

Chapter 8: FinOps Practices

Making cost optimization a continuous process

Tagging Strategy

Without consistent resource tagging, you cannot identify which team, project, or environment is responsible for cost. Tagging is the foundation of all FinOps work. Enforce tags at resource creation via policy (AWS SCP, Azure Policy, GCP Organization Policy).

Tag Key	Example Values	Purpose
Environment	production, staging, dev, test	Separate prod cost from non-prod
Project	website-v2, api-backend, data-pipeline	Attribute cost to project
Owner	team-backend, alice@company.com	Identify who manages the resource
CostCenter	engineering, marketing, data	Chargeback/showback to departments
AutoShutdown	true, false	Flag non-prod for scheduled shutdown
CreatedDate	2026-01-15	Identify old/forgotten resources

The Weekly Cost Review

- Set a weekly 30-minute calendar event: review cost by service vs prior week
- Investigate any service line with over 10% week-over-week increase
- Check for new untagged resources created in the past 7 days
- Review Advisor/Recommender/Cost Explorer recommendations — act on high-confidence items
- Track cumulative savings from optimization actions in a simple spreadsheet
- Share monthly cost summary with engineering team — visibility drives accountability

Quick Wins Checklist

■ **Delete unattached EBS/PD/disks**	Immediate saving — pure waste
■ **Release unused static IPs**	~$3-7/mo per IP
■ **Delete old snapshots (> 90 days)**	Often 5-15% of storage bill
■ **Stop dev/test instances on schedule**	60-70% reduction on those instances
■ **Enable S3 Intelligent-Tiering**	Auto-saves on infrequent access data
■ **Purchase 1-year Reserved Instances**	35-40% on stable compute
■ **Switch gp2 EBS to gp3**	20% cheaper, same or better performance

■ **Switch to ARM/Graviton instances**	20% cheaper, often faster
■ **Enable CloudFront for S3 content**	75-90% egress reduction
■ **Tag all untagged resources**	Unlocks cost attribution
■ **Set billing alerts at 80% of budget**	Catch surprises before month end
■ **Review Trusted Advisor/Advisor weekly**	Free recommendations worth acting on

hostitworks.com

24 Free Web Hosting & Developer Tools

Free cloud, hosting, DNS, security and performance tools to help you manage your infrastructure.

■■ Hosting & Cloud

- Website Hosting Checker
- Hosting Cost Calculator
- AWS Cost Calculator
- Hosting Price Comparator

■ DNS & Network

- DNS Lookup Tool
- DNS Propagation Checker
- IP Lookup & WHOIS
- Port Checker

■ Security

- SSL Certificate Checker
- HTTP Header Checker
- Password Generator

→ Visit hostitworks.com for All 24 Free Tools ←

No account required. No signup. Just free tools that work.