AWS Cost Optimization Checklist

Your guide to immediate cloud savings

✓ **15 immediate cost-saving opportunities**

✓ **Rightsizing calculator templates**

✓ **Security cost optimization tips**

Introduction

Managing costs in AWS is a continuous challenge. This checklist was created to help technology leaders and engineers quickly identify and eliminate the most common sources of unnecessary expenses. Use it as a practical tool to conduct an internal audit and regain control over your cloud budget.

Part 1: 15 Immediate Cost-Saving Opportunities

Check the boxes to track your progress. These actions are the "low-hanging fruit" that often yield savings of 30% or more.

Compute Resources (EC2 & Compute)

- [ ] **1. Terminate unused EC2 instances:** Identify instances in CloudWatch with an average CPU usage below 5% over the last 30 days and shut them down.

- [ ] **2. Eliminate "zombie" Elastic IPs:** Delete all Elastic IP addresses that are not assigned to any running EC2 instance to avoid charges for unused resources.
- [ ] **3. Implement schedules for dev/test environments:** Automatically stop development and test instances outside of business hours (e.g., at 6 PM) and start them in the morning (e.g., at 8 AM).
- [ ] **4. Switch to Graviton-based instances:** For compatible workloads, changing to instances based on ARM processors (Graviton) can provide a better price-performance ratio.
- [ ] **5. Utilize Spot Instances for non-critical tasks:** Use Spot Instances for batch jobs, test environments, or CI/CD pipelines to achieve savings of up to 90% compared to on-demand prices.

Storage

- [ ] **6. Delete unattached EBS volumes:** EBS volumes incur costs even when they are not attached to any instance. Identify and delete them.
- [ ] **7. Change EBS volume type from gp2 to gp3:** gp3 volumes offer better performance and are up to 20% cheaper than their older gp2 counterparts.
- [ ] **8. Clean up obsolete EBS snapshots:** Define a retention policy and delete old snapshots that are no longer needed for data recovery.
- [ ] **9. Implement S3 lifecycle policies:** Automatically move less frequently accessed data to cheaper storage classes, such as S3 Infrequent Access or Glacier.[4]

Databases (RDS)

- [ ] **10. Stop or delete unused RDS instances:** Identify database instances that have had no connections in recent weeks.
- [ ] **11. Check RDS disk types:** Ensure you are not overusing expensive io1/io2 disks where more cost-effective gp3 disks would suffice.

Networking & Other Services

- [ ] **12. Reduce CloudWatch log retention:** The default "never expire" setting can generate significant costs. Set a reasonable retention period (e.g., 30-90 days) or move logs to S3.
- [ ] **13. Use VPC Gateway Endpoints for S3:** To avoid data processing costs through a NAT Gateway, configure a free endpoint for S3 in your VPC.
- [ ] **14. Consolidate accounts in AWS Organizations:** If you manage multiple accounts, consolidating them can provide access to volume discounts.
- [ ] **15. Set up budget alerts in AWS Budgets:** Configure alerts that will notify you when spending approaches a defined threshold (e.g., 80% of the monthly budget).

Part 2: Rightsizing Calculator Templates

Rightsizing is the process of matching resources to actual demand. Instead of complex calculators, use the framework and rules below to make informed decisions. Analyze data from at least 15-30 days to get a reliable picture.

Step 1: Collect Key Metrics

Use Amazon CloudWatch (with the agent enabled for memory) or third-party tools to collect the following data for each EC2/RDS instance over a 30-day period:

Metric	Tool	What it measures
Maximum CPU Utilization	CloudWatch	Peak processor usage
Maximum Memory Utilization	CloudWatch Agent	Peak RAM usage
Network In/Out	CloudWatch	Peak network traffic
Disk Read/Write Ops	CloudWatch	Disk operation load

Step 2: Apply Decision Rules

Use the table below as a template for making rightsizing decisions.

If...	And...	Then...	Example
Max CPU Utilization < 40%	**Max Memory Utilization < 40%**	**Safely downsize the instance by half** within the same family (e.g., from m5.2xlarge to m5.xlarge).[7]	A c5.4xlarge instance with peak CPU at 35% and memory at 30% can be changed to c5.2xlarge.
Max CPU Utilization > 80%	**Max Memory Utilization < 40%**	**Change to a CPU-optimized instance** (e.g., from the m5 family	An m5.xlarge instance with peak CPU at 90% and memory at 25% will

		to c5).	perform better as a c5.xlarge.
Max CPU Utilization < 40%	**Max Memory Utilization > 80%**	**Change to a memory-optimized instance** (e.g., from the m5 family to r5).	An m5.xlarge instance with peak CPU at 20% and memory at 95% will perform better as an r5.large.
Workload is variable and "bursty"	**Average CPU is low**	**Change to a "burstable" instance type** (T3 or T4g family).[2]	A development server that is idle most of the time but needs power for compilation.

Part 3: Security Cost Optimization Tips

Good security practices often go hand-in-hand with savings. Implementing security in a cost-conscious manner is key.

- **1. Use VPC Endpoints instead of NAT Gateway:** Accessing AWS services (like S3, DynamoDB) via a VPC Endpoint is not only more secure (traffic does not leave the AWS network) but also significantly cheaper, as it eliminates data processing fees from the NAT Gateway.
- **2. Automate regulatory compliance:** Use services like AWS Config and AWS Security Hub to continuously monitor configurations for compliance. Automation is cheaper and less error-prone than manual audits, and it helps avoid costly fines for violations.
- **3. Implement the principle of least privilege in IAM:** Precise permission management prevents the accidental or malicious launch of unauthorized, expensive resources. This is a fundamental security principle that directly impacts cost control.
- **4. Optimize data retention with security in mind:** Store only the data that is absolutely necessary for business or legal reasons. Data minimization is a security principle that simultaneously reduces storage costs and decreases the attack surface.
- **5. Utilize AWS Trusted Advisor:** This service, in addition to providing cost-saving recommendations, also offers security checks. It often points out publicly accessible resources that not only pose a risk but can also generate unexpected data transfer costs if abused.