

Here is a list of cost-saving methods, categorised by their potential impact, ease of implementation, and long-term value. The methods that provide immediate and significant cost reductions are listed first, followed by those that may require more strategic planning or take longer to deliver measurable savings.

1 Turn off non-production resources out of hours

Impact: High impact, moderate effort (identifying idle resources might require monitoring tools).
Ease: **AWS Trusted Advisor** or **CloudWatch** metrics can help identify underutilized resources.

Impact: High impact, low effort.
Ease: Easy to automate using **AWS Instance Scheduler**, scripts with **CloudWatch** events or using an external tool like Cloud Custodian.

2 Remove idle resources

3 Remove unused EBS volumes

Impact: Medium impact, especially for environments that use EBS snapshots for backups.
Ease: Can be automated using **Lambda** functions or **AWS Backup** lifecycle policies.

Impact: Medium impact, immediate results.
Ease: Can be automated with scripts that check for unattached volumes.

4 Delete old snapshots

5 Use Spot Instances

Impact: High impact with potentially significant savings.
Ease: Tools like **AWS Compute Optimizer** can help recommend instance types based on historical usage.

Impact: High impact when used for the right workloads.
Ease: Requires adjusting workloads to handle interruptions, but AWS offers tools like **EC2 Auto Scaling** with **Spot Fleets** to simplify this.

6 Right-size EC2 instances

7 Use S3 Intelligent Tiering

Impact: Medium impact for environments with large data stored in **EFS**.
Ease: Requires configuration of **lifecycle policies** but is relatively straightforward.

Impact: Medium to high impact for large datasets.
Ease: Easy to implement as it requires changing the storage class of existing S3 objects.

8 Use EFS Lifecycle Management

9 Ensure AWS Backup Lifecycles

Impact: Medium impact, especially for environments with extensive backup requirements.
Ease: Manageable through **AWS Backup** settings and **lifecycle policies**.

Impact: Medium impact, helps prevent unnecessary long-term storage costs.
Ease: Can be automated through **AWS Backup Policies**.

10 Clean-up recovery points

11 Tear down environments when not require

Impact: High impact for long-term, stable workloads.
Ease: Requires commitment and planning but is relatively straightforward to implement.

Impact: High impact when applicable.
Ease: Requires robust **automation** and **provisioning scripts** to recreate environments when needed.

12 Savings Plans and Reserved Instances

13 Shift suitable workloads to Lambda and/or Fargate

Impact: Medium impact for organisations with large compute and storage footprints.
Ease: May require testing to ensure compatibility before upgrading.

Impact: Medium to High impact, but high flexibility.
Ease: Requires refactoring to serverless or containerised models. Use **Lambda** for short tasks, **Fargate** for longer or more complex ones.

14 Use latest EC2 and EBS types

15 Remove unused buckets and objects

Impact: Low impact, but good for housekeeping.
Ease: Can be managed with S3 Storage Lense.

Impact: Medium to low impact depending on the amount of unused data.
Ease: Can be automated with **S3 lifecycle policies** or scripts.

16 Remove S3 failed multi-part uploads

17 Close Redundant AWS Accounts

Impact: Varies greatly depending on the number of redundant accounts and associated costs.
Ease: May require analysis of account usage and impact assessment before closing accounts.

AWS Cost Management Tools

- AWS Cost Optimization Hub
- AWS Compute Optimizer
- AWS Trusted Advisor
- Amazon CloudWatch (Cost Anomaly Detection)
- AWS Cost Explorer



<https://seconyx.com>



<https://www.linkedin.com/company/seconyx>



<https://x.com/seconyx>