Amazon Web Services (AWS) Cost Optimization Checklist

90-Day Implementation Guide for 40-70% Cost Reduction

Version: 1.0
Last Updated: January 2024
Author: CloudCostChefs
Target Audience: AWS administrators, cloud engineers, FinOps practitioners

Executive Summary

This comprehensive 90-day checklist provides a structured approach to implementing Amazon Web Services (AWS) cost optimization strategies. By following this systematic methodology, organizations can achieve 40-70% cost reduction while maintaining performance and reliability. The checklist leverages AWS's extensive service portfolio, advanced pricing models including Reserved Instances and Spot Instances, and native optimization tools to deliver maximum cost efficiency.

Expected Outcomes by Phase

- **Phase 1 (Days 1-30):** 20-40% cost reduction through foundational optimizations and quick wins
- **Phase 2 (Days 31-60):** 30-55% cost reduction through strategic commitments and advanced optimization
- **Phase 3 (Days 61-90):** 40-70% cost reduction through AI-driven optimization and cultural transformation

Phase 1: Foundation & Quick Wins (Days 1-30)

Target Savings: 20-40%

Week 1: Assessment & Visibility Setup

Day 1-2: Initial AWS Environment Assessment

- [ ] **Complete AWS Account Audit**
- Document all AWS accounts and their purposes (production, staging, development, sandbox)
- Identify account owners and stakeholders across the organization
- Map organizational structure to AWS account hierarchy and billing relationships
- Assess current AWS Organizations setup and consolidated billing configuration
- [ ] **Establish Cost Baseline Metrics**
- Export last 90 days of billing data from AWS Cost Explorer
- Calculate current monthly spend by account, service, and region
- Identify top 20 cost-driving services and resources across all accounts
- Document current resource utilization patterns and peak usage times
- [ ] **Set Up AWS Cost Explorer and Billing Dashboard**
- Configure Cost Explorer with appropriate access permissions for relevant team members
- Enable detailed billing reports and cost allocation tags
- Set up cost and usage reports (CUR) for granular analysis
- Configure billing alerts and budget notifications for proactive monitoring

Day 3-4: Monitoring & Alerting Foundation

- [ ] **Deploy AWS CloudWatch Cost Monitoring**
- Enable CloudWatch billing metrics across all regions and accounts
- Set up custom CloudWatch dashboards for cost visualization and tracking

- Configure CloudWatch alarms for unusual spending patterns and threshold breaches
- Establish baseline performance metrics for correlation with cost data
- [ ] **Implement AWS Budgets and Cost Controls**
- Create account-level budgets based on historical usage patterns and business requirements
- Set up budget alerts at 50%, 75%, 90%, and 100% thresholds with appropriate escalation
- Configure email and SNS notifications for budget overruns and anomalies
- Establish escalation procedures and approval workflows for budget violations
- [ ] **Enable AWS Cost Anomaly Detection**
- Activate Cost Anomaly Detection service for automated unusual spending identification
- Configure anomaly detection monitors for different services and cost dimensions
- Set up notification preferences for anomaly alerts and investigation workflows
- Establish procedures for anomaly investigation and root cause analysis

Day 5-7: Resource Inventory & Tagging Strategy

- [ ] **Complete Comprehensive Resource Inventory**
- Use AWS Config and AWS Systems Manager Inventory to catalog all resources
- Document resource types, sizes, utilization patterns, and business purposes
- Identify orphaned, unused, and underutilized resources across all accounts
- Map resources to business applications, owners, and cost centers for accountability
- [ ] **Implement Comprehensive Tagging Strategy**
- Define mandatory tags: Environment, Owner, CostCenter, Application, Project, BusinessUnit, CreatedBy, Purpose

- Create and enforce tagging policies using AWS Organizations Service Control Policies (SCPs)
- Apply tags to existing resources using AWS Resource Groups Tagging API and bulk operations
- Set up automated tagging for new resources using AWS Config Rules and Lambda functions
- [ ] **Establish Governance Framework**
- Create IAM policies and roles for cost management access and responsibilities
- Define resource creation approval workflows and spending limits by role
- Implement account-level and service-level resource quotas and limits
- Set up AWS CloudTrail for audit logging of cost-related activities and changes

Week 2: Quick Optimization Wins

Day 8-10: EC2 Compute Optimization

- [ ] **Implement EC2 Right-Sizing**
- Analyze EC2 instance utilization using CloudWatch metrics and AWS Compute Optimizer
- Identify oversized instances with consistently low CPU, memory, or network utilization
- Create right-sizing recommendations with projected cost savings and performance impact
- Implement gradual right-sizing starting with development and testing environments
- [ ] **Deploy Spot Instances for Suitable Workloads**
- Identify fault-tolerant workloads suitable for Spot Instance deployment
- Convert batch processing, CI/CD, and development workloads to Spot Instances
- Implement Spot Fleet configurations for automatic instance type diversification
- Set up Spot Instance interruption handling and automatic restart mechanisms

- [ ] **Optimize EC2 Instance Scheduling**
- Identify instances that can be stopped during off-hours (development, testing, training environments)
- Implement automated start/stop schedules using AWS Lambda and CloudWatch Events
- Configure instance hibernation for Windows instances to reduce startup time
- Set up monitoring and alerting for scheduled instance operations

Day 11-12: Storage Optimization

- [ ] **Implement S3 Storage Class Optimization**
- Analyze S3 access patterns using S3 Storage Class Analysis and S3 Intelligent-Tiering
- Create lifecycle policies to automatically transition objects to cheaper storage classes
- Move infrequently accessed data to S3 Standard-IA, One Zone-IA, Glacier, or Deep Archive
- Implement S3 Intelligent-Tiering for automatic cost optimization without performance impact
- [ ] **Optimize EBS Volume Configuration**
- Identify and delete unattached EBS volumes and orphaned snapshots
- Resize oversized EBS volumes based on actual usage patterns and growth projections
- Convert gp2 volumes to gp3 for better price-performance ratio
- Implement EBS snapshot lifecycle policies for automated backup retention management
- [ ] **Clean Up Unused Storage Resources**
- Delete unused S3 buckets and objects that are no longer needed
- Remove orphaned Elastic File System (EFS) and FSx file systems
- Clean up unused Amazon Machine Images (AMIs) and associated snapshots

- Implement automated cleanup policies for temporary and development storage

Day 13-14: Network & Data Transfer Optimization

- [ ] **Optimize Data Transfer Costs**
- Analyze data transfer patterns using VPC Flow Logs and CloudWatch metrics
- Implement regional resource placement strategies to minimize cross-region transfers
- Optimize load balancer configurations and eliminate unnecessary data movement
- Use VPC endpoints to reduce NAT Gateway costs for AWS service communications
- [ ] **Implement CloudFront CDN Optimization**
- Deploy CloudFront distributions for static content delivery and API acceleration
- Implement intelligent caching strategies to reduce origin requests and data transfer
- Configure appropriate cache behaviors and TTL settings for different content types
- Optimize CloudFront pricing class selection based on global user distribution

Week 3: Database & Application Optimization

Day 15-17: RDS and Database Optimization

- [ ] **Analyze RDS Usage Patterns and Costs**
- Review RDS instance utilization using Performance Insights and CloudWatch metrics
- Identify oversized database instances and opportunities for right-sizing
- Analyze database storage usage patterns and growth trends
- Assess current backup and snapshot retention policies for optimization opportunities
- [ ] **Implement RDS Cost Optimization**
- Right-size RDS instances based on actual CPU, memory, and I/O utilization

- Convert appropriate workloads to Aurora Serverless for variable usage patterns
- Implement RDS Reserved Instances for predictable database workloads
- Optimize storage types (gp2 to gp3) and allocated storage based on usage
- [ ] **Optimize Database Backup and Retention**
- Review and optimize automated backup retention periods based on business requirements
- Implement cross-region backup strategies only where necessary for compliance
- Use database snapshots efficiently and clean up unnecessary manual snapshots
- Consider Aurora Global Database for disaster recovery instead of cross-region replication

Day 18-19: Application and Container Optimization

- [ ] **Optimize Lambda Function Costs**
- Analyze Lambda function execution patterns, duration, and memory usage
- Right-size Lambda function memory allocation based on actual requirements
- Implement Lambda Provisioned Concurrency only where necessary for performance
- Optimize Lambda function code for faster execution and reduced costs
- [ ] **Optimize ECS and EKS Container Costs**
- Analyze container resource utilization and right-size CPU and memory allocations
- Implement Fargate Spot for fault-tolerant container workloads
- Use EC2 Spot Instances for EKS worker nodes where appropriate
- Optimize container image sizes and implement multi-stage builds for efficiency

Day 20-21: Additional Service Optimization

- [ ] **Optimize CloudWatch Costs**
- Review CloudWatch Logs retention periods and adjust based on compliance requirements
- Implement log filtering and sampling to reduce ingestion costs

- Optimize custom metrics usage and eliminate unnecessary detailed monitoring
- Use CloudWatch Logs Insights efficiently for log analysis and troubleshooting
- [ ] **Optimize Other AWS Services**
- Review API Gateway usage and implement caching where appropriate
- Optimize Elasticsearch/OpenSearch cluster sizing and instance types
- Review and optimize SNS, SQS, and other messaging service usage
- Implement efficient backup strategies for various AWS services

Week 4: Governance & Automation Setup

Day 22-24: Advanced Governance Implementation

- [ ] **Implement Service Control Policies (SCPs)**
- Create SCPs to restrict expensive instance types in development accounts
- Implement region restrictions to control costs and ensure compliance
- Set up SCPs to enforce tagging requirements and prevent untagged resource creation
- Configure SCPs to require approval for high-cost resource creation
- [ ] **Deploy Advanced IAM Cost Controls**
- Implement condition-based IAM policies that restrict actions based on cost thresholds
- Create IAM roles with spending limits and approval requirements
- Set up cross-account roles for centralized cost management and monitoring
- Implement least-privilege access principles for cost-sensitive operations
- [ ] **Set Up AWS Config for Compliance**
- Deploy AWS Config rules for cost optimization compliance monitoring
- Implement rules for required tagging, approved instance types, and resource limits
- Set up automated remediation for common compliance violations

- Configure Config aggregators for multi-account compliance reporting

Day 25-27: Automation Framework Implementation

- [ ] **Deploy Cost Monitoring Automation**
- Set up automated cost reporting using Lambda functions and CloudWatch Events
- Implement automated anomaly detection and alerting workflows
- Create automated budget reconciliation and variance analysis
- Deploy cost optimization recommendation automation using AWS APIs
- [ ] **Implement Resource Lifecycle Automation**
- Set up automated resource cleanup using Lambda functions and CloudWatch Events
- Implement scheduled start/stop automation for development and testing resources
- Create automated scaling policies based on usage patterns and cost thresholds
- Deploy automated tagging and compliance remediation workflows

Day 28-30: Phase 1 Review & Optimization

- [ ] **Measure Phase 1 Results and Impact**
- Calculate actual cost savings achieved during the first 30 days
- Analyze resource utilization improvements and efficiency gains
- Review governance compliance metrics and policy effectiveness
- Document lessons learned, best practices, and areas for improvement
- [ ] **Prepare for Phase 2 Strategic Optimization**
- Identify additional optimization opportunities for Reserved Instances and Savings Plans
- Plan for advanced service-specific optimizations and architectural improvements
- Prepare for commitment-based pricing model implementations
- Set up advanced monitoring and analytics for Phase 2 activities

---

Phase 2: Strategic Optimization (Days 31-60)

Target Savings: 30-55%

Week 5: Reserved Instances & Savings Plans Implementation

Day 31-33: Reserved Instance Analysis and Planning

- [ ] **Analyze Reserved Instance Opportunities**
- Review historical EC2, RDS, and other service usage patterns for commitment suitability
- Calculate potential savings from Reserved Instances across different terms and payment options
- Identify stable workloads suitable for 1-year and 3-year Reserved Instance commitments
- Analyze regional vs. availability zone Reserved Instance options for flexibility
- [ ] **Implement Reserved Instance Strategy**
- Purchase Reserved Instances for predictable EC2 workloads with highest savings potential
- Implement RDS Reserved Instances for stable database workloads
- Consider ElastiCache, Redshift, and other service-specific Reserved Instance options
- Set up Reserved Instance utilization monitoring and optimization alerts
- [ ] **Optimize Reserved Instance Management**
- Implement Reserved Instance modification and exchange strategies for changing requirements
- Set up automated Reserved Instance utilization reporting and recommendations
- Configure alerts for underutilized Reserved Instances and optimization opportunities
- Plan for Reserved Instance renewal and capacity planning processes

Day 34-36: Savings Plans Implementation

- [ ] **Evaluate Savings Plans Options**
- Analyze Compute Savings Plans vs. EC2 Instance Savings Plans for different workloads
- Calculate potential savings and flexibility trade-offs for different commitment levels
- Assess Savings Plans coverage for Lambda, Fargate, and other compute services
- Plan Savings Plans implementation strategy with gradual commitment increases
- [ ] **Implement Savings Plans Strategy**
- Purchase Compute Savings Plans for flexible workloads with variable instance types
- Implement EC2 Instance Savings Plans for predictable workloads with specific instance families
- Set up Savings Plans utilization monitoring and optimization tracking
- Configure automated recommendations for additional Savings Plans opportunities
- [ ] **Optimize Commitment-Based Pricing**
- Balance Reserved Instances and Savings Plans for optimal cost efficiency and flexibility
- Implement hybrid pricing strategies combining On-Demand, Reserved, Spot, and Savings Plans
- Set up automated commitment utilization optimization and rebalancing
- Monitor and optimize commitment coverage across different services and workloads

Day 37-38: Advanced Spot Instance Integration

- [ ] **Expand Spot Instance Usage**
- Identify additional workloads suitable for Spot Instance deployment

- Implement Spot Fleet configurations with multiple instance types and availability zones
- Deploy Spot Instances for batch processing, data analysis, and CI/CD workloads
- Set up automated Spot Instance bidding strategies and price optimization
- [ ] **Implement Spot Instance Best Practices**
- Configure Spot Instance interruption handling and graceful shutdown procedures
- Implement checkpointing and state management for long-running Spot workloads
- Set up Spot Instance diversification across instance types and availability zones
- Monitor Spot Instance savings and optimization opportunities continuously

Week 6: Advanced Service Optimization

Day 39-41: Advanced Storage Optimization

- [ ] **Implement Advanced S3 Optimization**
- Deploy S3 Transfer Acceleration for global data uploads and downloads
- Implement S3 Cross-Region Replication optimization for disaster recovery
- Use S3 Batch Operations for large-scale object management and optimization
- Optimize S3 request patterns and implement efficient data access strategies
- [ ] **Advanced EBS and Storage Optimization**
- Implement EBS Multi-Attach for shared storage scenarios where appropriate
- Optimize EBS snapshot strategies and implement cross-region backup policies
- Use EFS Intelligent-Tiering for automatic file system cost optimization
- Implement FSx optimization for high-performance workloads

Day 42-44: Advanced Database and Analytics Optimization

- [ ] **Implement Advanced RDS Optimization**
- Deploy Aurora Global Database for global applications with local read performance

- Implement Aurora Serverless v2 for variable database workloads
- Optimize RDS Proxy usage for connection pooling and improved efficiency
- Use RDS Performance Insights for query optimization and cost reduction
- [ ] **Optimize Data Analytics Services**
- Implement Redshift Reserved Nodes for predictable data warehouse workloads
- Use Redshift Spectrum for cost-effective querying of S3 data
- Optimize EMR cluster configurations and implement Spot Instance usage
- Implement Athena query optimization and result caching strategies

Day 45-46: Container and Serverless Optimization

- [ ] **Advanced Container Optimization**
- Implement EKS cluster autoscaling and node group optimization
- Use Karpenter for intelligent node provisioning and cost optimization
- Optimize container resource requests and limits for efficient packing
- Implement multi-architecture container images for ARM-based instances
- [ ] **Advanced Serverless Optimization**
- Optimize Lambda function performance and cost through profiling and tuning
- Implement Lambda Extensions for shared initialization and caching
- Use Step Functions for complex workflow orchestration and cost optimization
- Optimize API Gateway usage and implement efficient caching strategies

Week 7: Network and Security Optimization

Day 47-49: Advanced Network Optimization

- [ ] **Implement Advanced VPC Optimization**
- Optimize VPC peering and Transit Gateway configurations for cost efficiency
- Implement VPC endpoints for all supported AWS services to reduce NAT costs
- Use AWS PrivateLink for secure and cost-effective service connectivity

- Optimize Direct Connect and VPN configurations for hybrid connectivity
- [ ] **Advanced Load Balancer Optimization**
- Right-size Application Load Balancers and Network Load Balancers based on traffic
- Implement efficient load balancer target group configurations
- Use Gateway Load Balancers for third-party appliance integration where appropriate
- Optimize load balancer idle timeout and connection settings

Day 50-52: Security and Compliance Optimization

- [ ] **Implement Cost-Effective Security**
- Optimize AWS WAF rules and request sampling for cost efficiency
- Use AWS Shield Advanced only where necessary for DDoS protection
- Implement efficient AWS Config rule deployment and evaluation
- Optimize CloudTrail logging and data event selection for compliance needs
- [ ] **Compliance and Governance Optimization**
- Implement efficient AWS Security Hub and GuardDuty configurations
- Optimize AWS Inspector and Systems Manager Patch Manager usage
- Use AWS Trusted Advisor recommendations for security and cost optimization
- Implement cost-effective backup and disaster recovery strategies

Week 8: Advanced Monitoring and Analytics

Day 53-55: Advanced CloudWatch Optimization

- [ ] **Implement Advanced CloudWatch Strategies**
- Optimize CloudWatch custom metrics and reduce unnecessary detailed monitoring
- Implement efficient CloudWatch Logs aggregation and filtering
- Use CloudWatch Synthetics efficiently for application monitoring

- Optimize CloudWatch Events and EventBridge usage for automation

- [ ] **Implement Advanced Cost Analytics**

- Set up advanced cost analytics using AWS Cost and Usage Reports in S3
- Create custom cost dashboards using QuickSight or third-party tools
- Implement predictive cost modeling using machine learning services
- Set up automated cost optimization reporting and recommendations

Day 56-58: Performance and Cost Correlation

- [ ] **Implement Performance Monitoring**
- Correlate application performance metrics with cost data for optimization insights
- Implement AWS X-Ray for distributed tracing and performance optimization
- Use Application Performance Monitoring (APM) tools for cost-performance analysis
- Optimize performance vs. cost trade-offs based on business requirements

Day 59-60: Phase 2 Review and Planning

- [ ] **Measure Phase 2 Results and Cumulative Impact**
- Calculate total cost savings achieved through strategic optimization efforts
- Analyze the effectiveness of Reserved Instances and Savings Plans implementations
- Review advanced optimization strategies and their impact on performance and costs
- Plan for Phase 3 advanced automation and cultural transformation initiatives

---

Phase 3: Advanced Optimization & Culture (Days 61-90)

Target Savings: 40-70%

Week 9: AI-Driven Optimization & Advanced Analytics

Day 61-63: Machine Learning-Based Optimization

- [ ] **Implement AWS Compute Optimizer Automation**
- Set up automated processing of Compute Optimizer recommendations
- Implement machine learning-based cost prediction models using SageMaker
- Use AWS Cost Anomaly Detection with custom machine learning models
- Deploy intelligent alerting and automated optimization workflows
- [ ] **Advanced Analytics and Forecasting**
- Implement predictive cost modeling using historical usage patterns
- Use AWS Forecast for demand planning and capacity optimization
- Deploy advanced analytics for usage pattern recognition and optimization
- Implement intelligent resource scheduling based on predicted demand

Day 64-66: Predictive Analytics & Intelligent Automation

- [ ] **Implement Cost Forecasting and Planning**
- Deploy predictive cost modeling for budget planning and variance analysis
- Implement automated budget forecasting based on historical trends and growth patterns
- Set up trend analysis and reporting for proactive cost management
- Create automated cost projections for different scenarios and business growth
- [ ] **Advanced Automation and Intelligence**
- Implement real-time cost optimization using AWS Lambda and machine learning
- Deploy advanced visualization and reporting using AWS QuickSight
- Set up automated insights and recommendations using AWS Personalize
- Implement intelligent cost optimization scoring and ranking systems

Week 10: Multi-Account & Organization Optimization

Day 67-69: Organization-Level Optimization

- [ ] **Implement Cross-Account Optimization Strategies**
- Optimize resource sharing across accounts using AWS Resource Access Manager
- Implement centralized cost management and optimization across AWS Organizations
- Deploy organization-level policies for cost control and governance
- Set up cross-account monitoring and reporting for comprehensive visibility
- [ ] **Advanced Resource Management and Sharing**
- Implement intelligent resource allocation across multiple accounts
- Deploy automated resource optimization workflows at the organization level
- Set up advanced quota management and cost allocation across business units
- Implement cost-aware resource scheduling and allocation policies

Day 70-72: Advanced Integration and Automation

- [ ] **Deploy Enterprise-Grade Automation**
- Implement intelligent resource lifecycle management across all accounts
- Set up automated cost optimization workflows with approval and rollback capabilities
- Deploy advanced monitoring and alerting with intelligent escalation
- Implement automated remediation for cost optimization opportunities
- [ ] **Integration and API Optimization**
- Optimize AWS API usage and implement efficient service integration patterns
- Implement cost-aware application design and architecture patterns
- Deploy automated cost optimization in CI/CD pipelines and development workflows
- Set up integration with enterprise tools and third-party cost management platforms

Week 11: FinOps Culture & Process Optimization

Day 73-75: FinOps Culture Implementation

- [ ] **Establish FinOps Practices and Culture**
- Train development and operations teams on cost optimization principles
- Implement cost awareness in software development and deployment processes
- Set up cost optimization incentives and accountability measures
- Create communities of practice for cost optimization knowledge sharing
- [ ] **Process Integration and Optimization**
- Integrate cost optimization into software development lifecycle (SDLC)
- Implement cost-aware architecture reviews and design processes
- Set up cost optimization considerations in project planning and budgeting
- Deploy cost optimization metrics and KPIs for team performance measurement

Day 76-78: Advanced Training & Enablement

- [ ] **Team Training and Certification Programs**
- Provide comprehensive AWS cost optimization training for technical teams
- Implement AWS certification programs focused on cost optimization
- Set up regular knowledge sharing sessions and best practice workshops
- Create comprehensive cost optimization documentation and runbooks
- [ ] **Tool and Process Optimization**
- Optimize cost management tools and processes for maximum efficiency
- Implement advanced reporting and analytics for different stakeholder groups
- Set up automated training and enablement programs for new team members
- Deploy cost optimization best practices and standards across the organization

Week 12: Continuous Improvement & Future Planning

Day 79-81: Continuous Improvement Framework

- [ ] **Implement Continuous Optimization Processes**
- Set up automated optimization cycles with regular review and improvement
- Implement continuous monitoring and improvement of cost optimization strategies
- Deploy advanced analytics and insights for ongoing optimization opportunities
- Set up automated optimization recommendations and implementation workflows
- [ ] **Advanced Governance & Compliance**
- Implement advanced governance frameworks for cost optimization and compliance
- Set up automated compliance monitoring and reporting for cost-related policies
- Deploy cost optimization policies and procedures across the organization
- Implement advanced audit and reporting capabilities for cost optimization activities

Day 82-84: Strategic Planning & Future Roadmap

- [ ] **Long-term Strategic Planning**
- Develop comprehensive long-term cost optimization strategy and roadmap
- Plan for future AWS service adoption and cost optimization opportunities
- Set up advanced cost optimization roadmap with measurable goals and milestones
- Implement strategic cost management aligned with business objectives and growth

Day 85-87: Documentation & Knowledge Transfer

- [ ] **Complete Documentation and Knowledge Management**
- Document all optimization implementations, procedures, and best practices

- Create comprehensive runbooks and operational procedures for ongoing management
- Set up knowledge transfer processes and documentation maintenance procedures
- Implement documentation version control and regular review processes

Day 88-90: Final Assessment & Celebration

- [ ] **Final Assessment and Results Reporting**
- Calculate total cost savings achieved throughout the 90-day implementation
- Analyze optimization effectiveness and return on investment for all initiatives
- Create comprehensive executive summary and presentation for stakeholders
- Plan for ongoing optimization and continuous improvement beyond the initial 90 days

---

Success Metrics & KPIs

Financial Metrics

- **Total Cost Reduction:** Target 40-70% reduction in monthly AWS spend
- **Cost per Workload:** Measure cost efficiency improvements per application or service
- **ROI on Optimization Efforts:** Calculate return on investment for optimization initiatives and tools
- **Budget Variance:** Track actual vs. budgeted spend across all accounts and services

Operational Metrics

- **Resource Utilization:** Target >80% average utilization across EC2 instances
- **Reserved Instance Utilization:** Target >95% utilization of purchased Reserved Instances

- **Savings Plans Coverage:** Target >70% of eligible compute spend covered by Savings Plans
- **Spot Instance Adoption:** Target >40% of suitable workloads running on Spot Instances

Governance Metrics

- **Tagging Compliance:** Target >95% of resources properly tagged according to policy
- **Policy Compliance:** Target >90% compliance with organizational cost control policies
- **Budget Alert Response Time:** Target <24 hours for budget threshold alert response
- **Cost Anomaly Detection:** Target <4 hours for anomaly identification and response

AWS-Specific Metrics

- **Service Optimization Coverage:** Target optimization implementation across >90% of used AWS services
- **Multi-Account Governance:** Target >95% compliance across all AWS accounts in organization
- **Automation Coverage:** Target >80% of optimization activities automated
- **Cost Allocation Accuracy:** Target >95% of costs properly allocated to business units

---

Risk Management & Mitigation

Common Risks and Mitigation Strategies

Performance Impact Risk

- **Risk:** Aggressive cost optimizations may negatively impact application performance

- **Mitigation:** Implement gradual changes with comprehensive performance monitoring and testing
- **Monitoring:** Set up performance baselines and continuous monitoring for all optimization changes
- **Rollback:** Maintain detailed rollback procedures for all optimization implementations

Service Availability Risk

- **Risk:** Cost cutting measures may impact service reliability and availability
- **Mitigation:** Implement optimization changes in non-production environments first
- **Testing:** Conduct thorough testing and validation before production implementation
- **Monitoring:** Set up comprehensive availability monitoring and alerting

Compliance and Security Risk

- **Risk:** Cost optimizations may conflict with compliance or security requirements
- **Mitigation:** Review all changes with compliance and security teams before implementation
- **Documentation:** Maintain detailed documentation of all changes and their compliance impact
- **Audit:** Implement regular compliance audits and security reviews

Team Adoption and Change Management Risk

- **Risk:** Teams may resist cost optimization changes or lack necessary skills
- **Mitigation:** Implement comprehensive training and change management programs
- **Communication:** Maintain clear communication about benefits, goals, and expectations
- **Incentives:** Create appropriate incentives and accountability measures for cost optimization

---

Tools & Resources

AWS Native Tools

- **AWS Cost Explorer:** Primary cost analysis and visualization tool
- **AWS Trusted Advisor:** Automated recommendations for cost optimization
- **AWS Compute Optimizer:** Machine learning-powered right-sizing recommendations
- **AWS Budgets:** Proactive cost control with custom budgets and alerts
- **AWS Cost Anomaly Detection:** Automated unusual spending pattern detection

Third-Party Tools

- **CloudCostChefs Multi-Cloud Tools:** Cross-cloud optimization and management
- **Mise-en-Tag Enforcer for AWS:** Automated tagging enforcement and governance
- **Mise-en-Place VM Scheduler:** Automated VM lifecycle management
- **Custom Analytics Dashboards:** Advanced cost analytics and reporting

Automation Scripts and Templates

- **AWS Cost Monitor:** Real-time cost monitoring and alerting automation
- **Resource Optimizer:** Automated resource right-sizing and optimization
- **Commitment Optimizer:** Automated Reserved Instance and Savings Plans optimization
- **Storage Lifecycle Manager:** Automated S3 and EBS optimization

---

Conclusion

This comprehensive 90-day AWS cost optimization checklist provides a structured approach to achieving significant cost reductions while maintaining performance and reliability. By following this systematic methodology and leveraging AWS's extensive service portfolio, advanced pricing models, and native optimization tools, organizations can achieve 40-70% cost reduction.

The key to success is consistent execution, continuous monitoring, and cultural adoption of cost optimization practices. Regular review and adjustment of optimization strategies ensure sustained cost efficiency and maximum return on cloud investment.

For additional support and advanced optimization strategies, consider leveraging CloudCostChefs' AWS-specific tools and expertise to accelerate your AWS cost optimization journey.

Document Version: 1.0
Last Updated: January 2024
Next Review: July 2024
Contact: CloudCostChefs Support Team