Save 30-50% on AWS in Under 5 Minutes: The Complete Setup Guide

Most articles about AWS cost optimization tell you to right-size instances, delete unused resources, and buy Reserved Instances. They assume you have unlimited time to manage commitments and the risk tolerance to lock in three-year contracts.

The reality is different. Your engineering team has products to build, not spreadsheets to manage. Your usage patterns change monthly. And every day you delay optimization costs you thousands in unnecessary spending.

We built Usage AI because we saw companies collectively wasting millions on AWS. Not from incompetence, but from the sheer complexity of managing cloud commitments. After helping customers save over $91 million, we've learned exactly what works and what doesn't.

Skip to the setup guide if you want to get started right away.

Why AWS Cost Optimization Fails

AWS offers incredible discounts through Savings Plans and Reserved Instances. Compute Savings Plans provide up to 66% discounts in exchange for a commitment to consistent usage for 1 or 3 year terms. The problem isn't the discounts, it's the commitment risk.

Consider what happens when you buy a three-year Reserved Instance. Your architecture evolves constantly as your business grows and technology advances, but those commitments remain frozen in time. You committed to m5.xlarge instances in us-east-1, but six months later, your team decides to migrate to m6i instances for better performance, or shift workloads to us-west-2 for latency reasons. Now you're stuck paying for unused reservations while also paying on-demand rates for your new infrastructure.

The situation gets worse when usage patterns shift. Maybe you optimized for your current load of 100 instances running 24/7. Then a major customer churns, or you refactor an inefficient service, and suddenly you need 60 instances. Those 40 unused reservations become dead money bleeding from your budget every month. Before 2024, you could sell unwanted RIs on the AWS Reserved Instance Marketplace, but AWS effectively killed that escape route by preventing Enterprise Discount Program customers from selling and limiting how many RIs any single customer can sell.

Most companies respond to this risk by staying on-demand, effectively paying a 50% “flexibility tax” every month. They know they're overpaying, but the fear of overcommitment paralysis keeps them from acting. It's a rational response to an irrational system.

The 5 Services Eating Your Budget (And How to Fix Them)

Your AWS spend concentrates in five core services. Understanding each one's optimization strategy is crucial for maximizing savings while minimizing risk. Let me walk you through each service, explaining not just what to do, but why these strategies work and what pitfalls to avoid.

1. EC2: The Compute Giant (40-60% of spend)

EC2 dominates most AWS bills because every application needs compute. Whether you're running web servers, application servers, batch processing jobs, or machine learning workloads, EC2 instances form the backbone of your infrastructure. The challenge is that EC2 offers hundreds of instance types across dozens of regions, each with different pricing, performance characteristics, and use cases.

Compute Savings Plans automatically apply to EC2 instance usage regardless of instance family,

size, AZ, Region, OS or tenancy, and also apply to Fargate or Lambda usage. This flexibility fundamentally changes the optimization equation. Instead of committing to specific instance types that might become obsolete or inappropriate for your workloads, you commit to a dollar amount of compute usage that follows you wherever your infrastructure evolves.

The smart strategy here is using 1-year Compute Savings Plans with no upfront payment. You get 27% discounts while maintaining complete flexibility to change instance types, regions, or even move to containers. This might seem like you're leaving money on the table compared to 3-year commitments with 54% discounts, but that assumes your infrastructure remains static for three years. In reality, AWS releases new instance families annually that offer better price-performance. The m6i instances released in 2021 are 15% more cost-effective than m5 instances. If you're locked into m5 Reserved Instances, you can't take advantage of these improvements.

For coverage targets, aim for 70-80% of your baseline usage. This provides substantial savings while maintaining surge capacity for traffic spikes or seasonal variations. If you spend $50,000 monthly on EC2, proper coverage could save you $13,500 monthly—that's $162,000 annually that can be reinvested into growth instead of padding AWS's margins.

2. RDS: The Database (15-25% of spend)

Databases present unique optimization challenges because they're the foundation of your application's data layer. Unlike stateless application servers that can be scaled up and down or moved between regions easily, databases carry state. They run 24/7 because even a few minutes of downtime can cascade into hours of recovery and reconciliation. This makes them perfect candidates for commitments—except nobody wants to lock in database configurations when data growth is unpredictable.

Think about your database growth pattern. You might have 10GB of data today, fitting comfortably on a db.m5.large instance. But if your business takes off, you could have 100GB in six months, requiring a db.m5.xlarge or larger. If you bought a Reserved Instance for the smaller size, you're now paying for an unused reservation while also paying on-demand rates for the larger instance. It's a double penalty that can cost thousands monthly.

The smart strategy for RDS focuses on Reserved Instances for production databases that haven't changed configurations in six months. These stable databases have predictable resource requirements and are unlikely to need sudden changes. Start with 60% coverage to maintain flexibility for growth or architectural changes. This conservative approach still yields 30-42% savings on one-year commitments while protecting you from overcommitment.

Keep development and staging databases on-demand. These environments need flexibility for testing different configurations, and their intermittent usage patterns make commitments

wasteful. Many companies make the mistake of treating all databases equally, but production and non-production databases have fundamentally different optimization profiles.

3. ElastiCache: The Hidden Optimizer (5-10% of spend)

ElastiCache often runs unnoticed in the background, quietly accumulating costs while teams focus on more visible services. Redis and Memcached clusters typically get configured once during application setup and then forgotten, running 24/7 at on-demand rates for years. This makes ElastiCache one of the easiest services to optimize with minimal risk.

Cache layers rarely change configuration once optimized because their sizing is determined by relatively stable factors: key size, number of keys, and access patterns. Unlike databases that grow with business data or compute instances that scale with traffic, cache requirements remain remarkably consistent. A Redis cluster sized for 10GB of hot data will likely still need 10GB six months later—the hot dataset size doesn't typically grow proportionally with overall data growth.

Reserved Instances for all production Redis and Memcached clusters represent one of the safest optimization moves you can make. The 35% savings from one-year RIs translate directly to your bottom line with virtually zero risk of overcommitment. The only caveat is that ElastiCache reservations are tied to node types, not clusters. If you need to scale from cache.m5.large to cache.m5.xlarge, your reservation doesn't automatically adjust. But this scenario is rare enough that the savings far outweigh the minimal risk.

4. Redshift: The Data Warehouse Drain (10-15% of spend)

Redshift poses interesting optimization challenges because data warehouses sit at the intersection of storage and compute. Your Redshift cluster needs enough storage for your data and enough compute for your queries, but these requirements don't always scale together. You might need more storage without additional compute, or more compute for complex queries without additional storage.

Redshift Reserved Nodes offer 41% savings even on one-year terms, but they're remarkably inflexible. You're locked to specific node types (dc2.large, dc2.8xlarge, ra3.4xlarge, etc.) in specific regions. There's no equivalent to Compute Savings Plans that let you shift between node types. If you commit to dc2 nodes and later need the storage flexibility of ra3 nodes, you're stuck with unused reservations.

The smart strategy acknowledges this inflexibility by only committing to clusters with stable, predictable workloads. Your core data warehouse that's been running the same node configuration for a year is a good candidate. The experimental cluster you're using to test new analytics workloads should stay on-demand. Coverage recommendations of 50-60% account for growth while avoiding overcommitment. This conservative approach still yields meaningful

savings—if you're spending $20,000 monthly on Redshift, even 50% coverage with RIs saves $4,100 monthly.

5. OpenSearch: The Overlooked Service (5-10% of spend)

OpenSearch, previously known as Elasticsearch, often escapes optimization efforts because it falls between teams. The operations team sees it as an application service, the application team sees it as infrastructure, and nobody takes ownership of its cost optimization. This orphaned status means OpenSearch clusters frequently run for years at on-demand rates despite being perfect candidates for Reserved Instances.

Search clusters, like cache clusters, have predictable resource requirements determined by index size and query patterns. Once you've sized a cluster for your search workload, it rarely needs dramatic changes. The data might grow, but search indices are typically time-bounded (last 90 days of logs, last year of transactions) or size-capped, creating natural limits on growth.

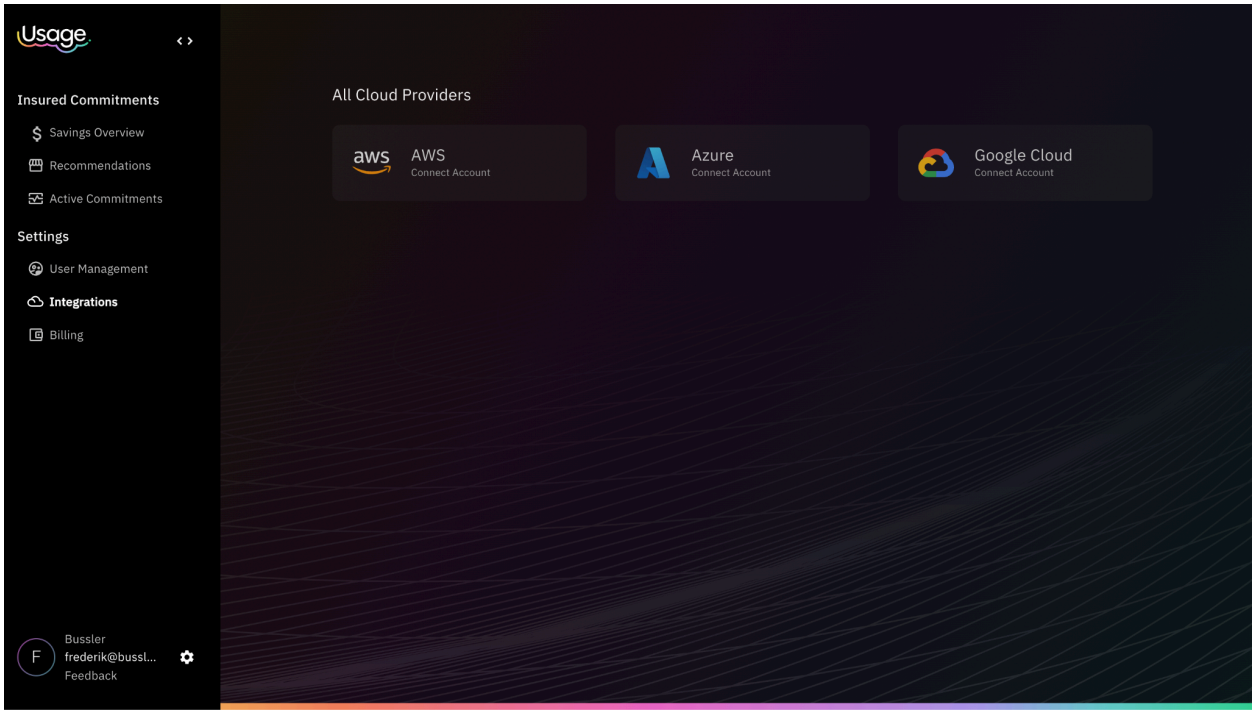
One-year Reserved Instances for production OpenSearch clusters offer 32% savings with minimal risk. Most companies can cover 100% of their OpenSearch spend with RIs because these clusters run continuously with stable configurations. The savings might seem smaller in percentage terms than EC2 or RDS, but OpenSearch often represents $10,000-50,000 in monthly spend for medium-sized companies. That's $3,200-16,000 in monthly savings from a single optimization decision that takes minutes to implement.

The Actual 5-Minute Setup Process

Here's exactly how to connect your AWS account to Usage.AI for a free savings analysis.

Step 1: Start the Connection (30 seconds)

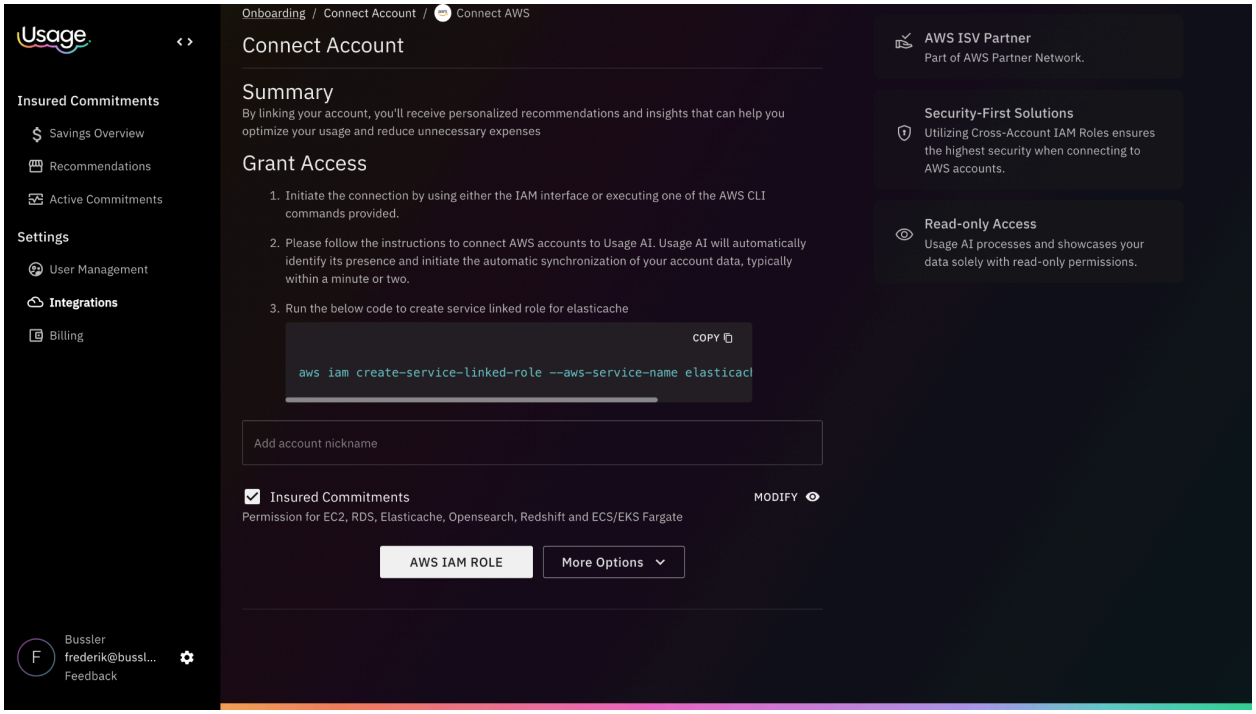
Navigate to usage.ai and create an account using your business email. We intentionally don't require credit cards or lengthy forms. Just email, password, and company name. Once you're in the dashboard, simply click AWS to begin connecting.



Step 2: Initial Configuration (1 minute)

The first screen asks for an account nickname. This is purely for your reference if you manage multiple AWS accounts. Something like "Production Account" or "Company-AWS-Main" works just fine. Below that, you'll see a product selection checkbox for "Insured Commitments." Make sure this is checked—it grants permissions for EC2, RDS, ElastiCache, OpenSearch, Redshift, and ECS/EKS Fargate. You can modify this selection later if needed, but starting with full permissions gives you the most comprehensive analysis.

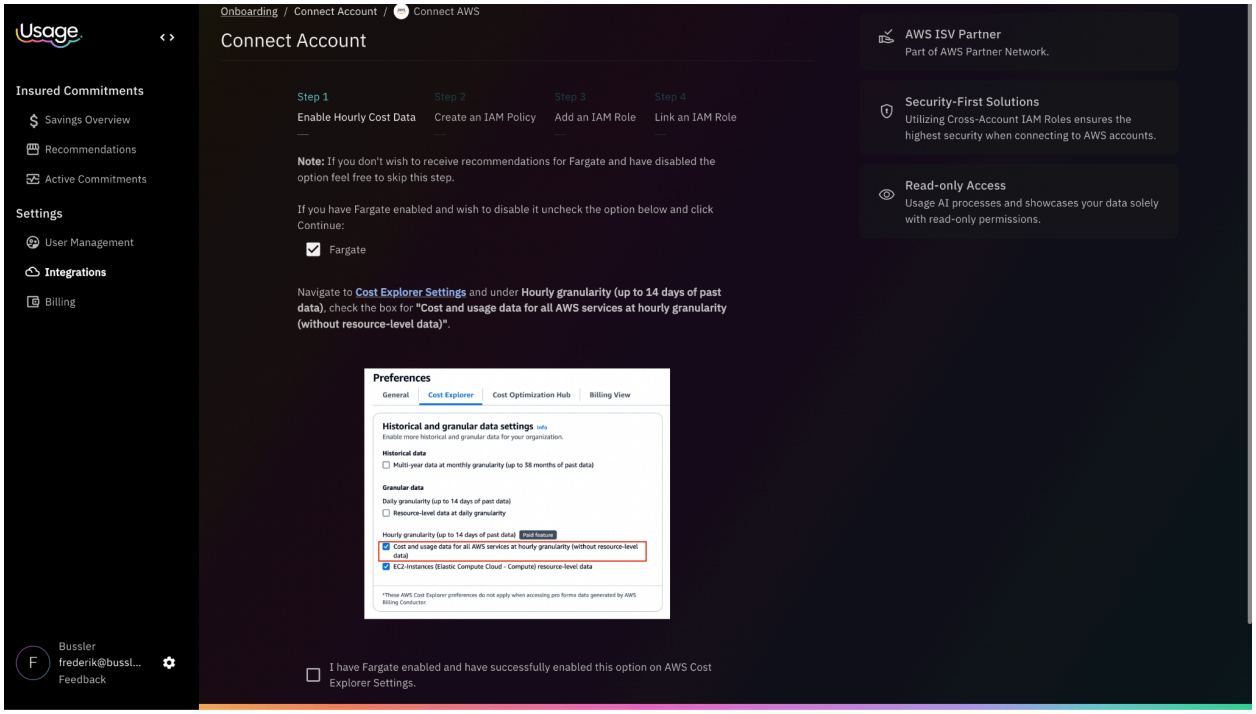
Then, copy the given code and run it to create a service linked role for elasticache.



Click the AWS IAM ROLE button to continue. This takes you to the permissions configuration screen where the real AWS integration begins.

Step 3: Enable Hourly Cost Data (30 seconds - skip if no Fargate)

If you use Fargate for containerized workloads, you need to enable hourly cost granularity in AWS Cost Explorer. This sounds complex but takes seconds. Open a new browser tab and navigate to your AWS Console. Go to Cost Explorer Settings (you can search for it in the top search bar). Look for the section labeled "Hourly Granularity" and check the box that says "Cost and usage data for all AWS services at hourly granularity (without resource-level data)."

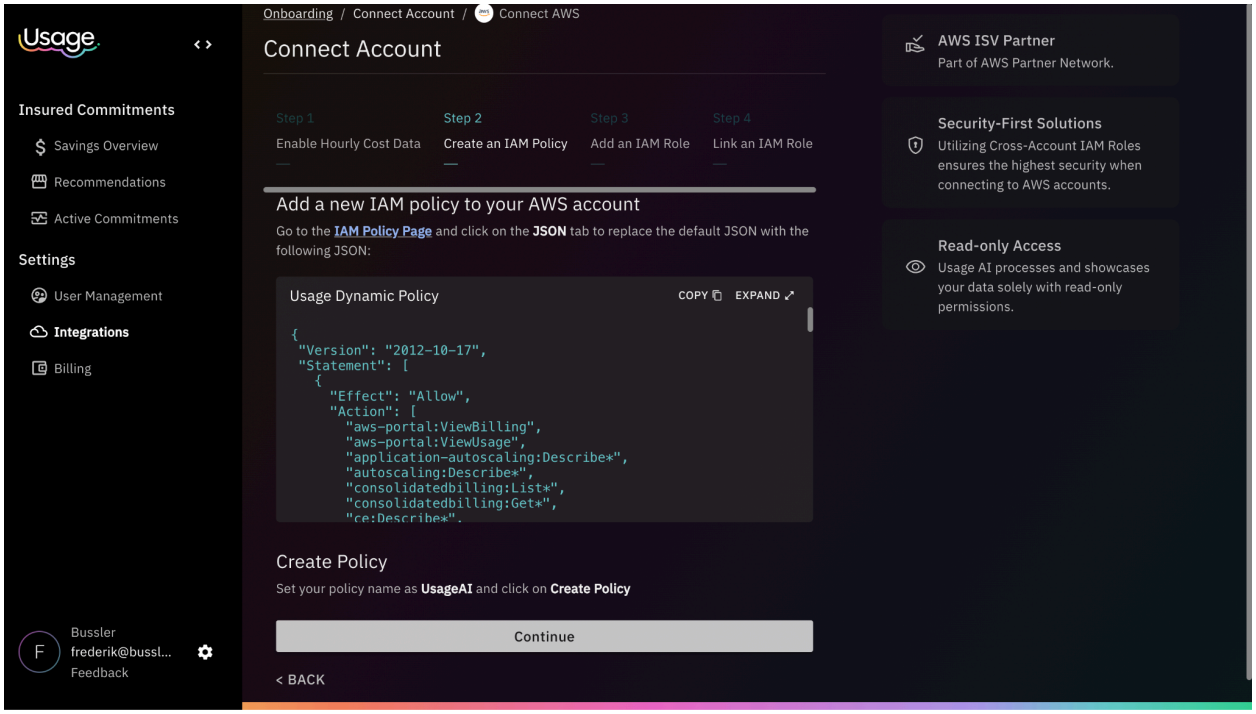


Return to the Usage.AI dashboard and check the confirmation box saying you've enabled this option. If you don't use Fargate, skip this entire step. The platform will still analyze your EC2, RDS, and other services without hourly data.

Step 4: Create the IAM Policy (1 minute)

Now we need to create the read-only policy that lets Usage.AI analyze your spending. In your AWS Console tab, navigate to the IAM service and click on Policies in the left sidebar. Click the blue "Create Policy" button. You'll see a visual editor by default—ignore it and click on the JSON tab instead.

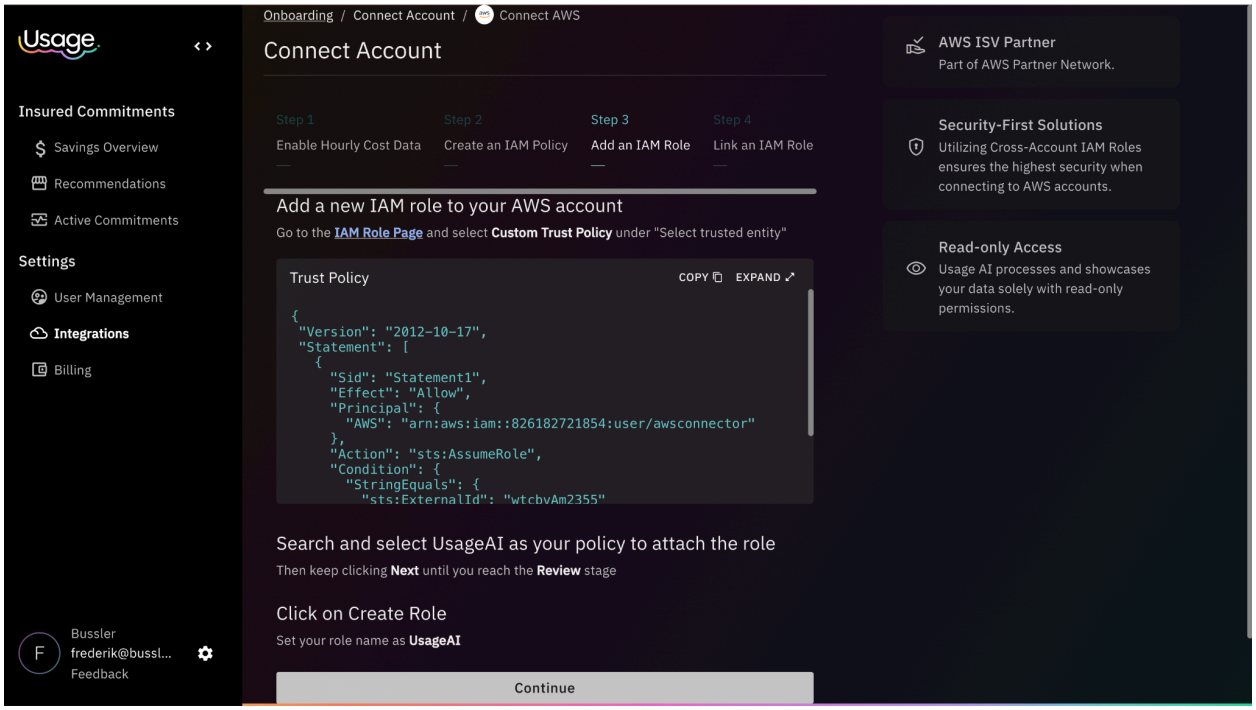
The Usage.AI dashboard displays a JSON policy.



Copy this entire policy from the Usage.AI dashboard and paste it into AWS, replacing any default content. Click through to the review screen, name the policy "UsageAI" (exactly as shown, without the quotation marks), and click Create Policy. The policy is now ready to be attached to a role.

Step 5: Create the IAM Role (1 minute)

Return to the IAM dashboard and click Roles in the left sidebar. Click "Create Role" and select "Custom Trust Policy" as the trusted entity type. The Usage.AI dashboard provides another JSON block—this is the trust policy that allows Usage.AI's AWS account to assume this role securely.

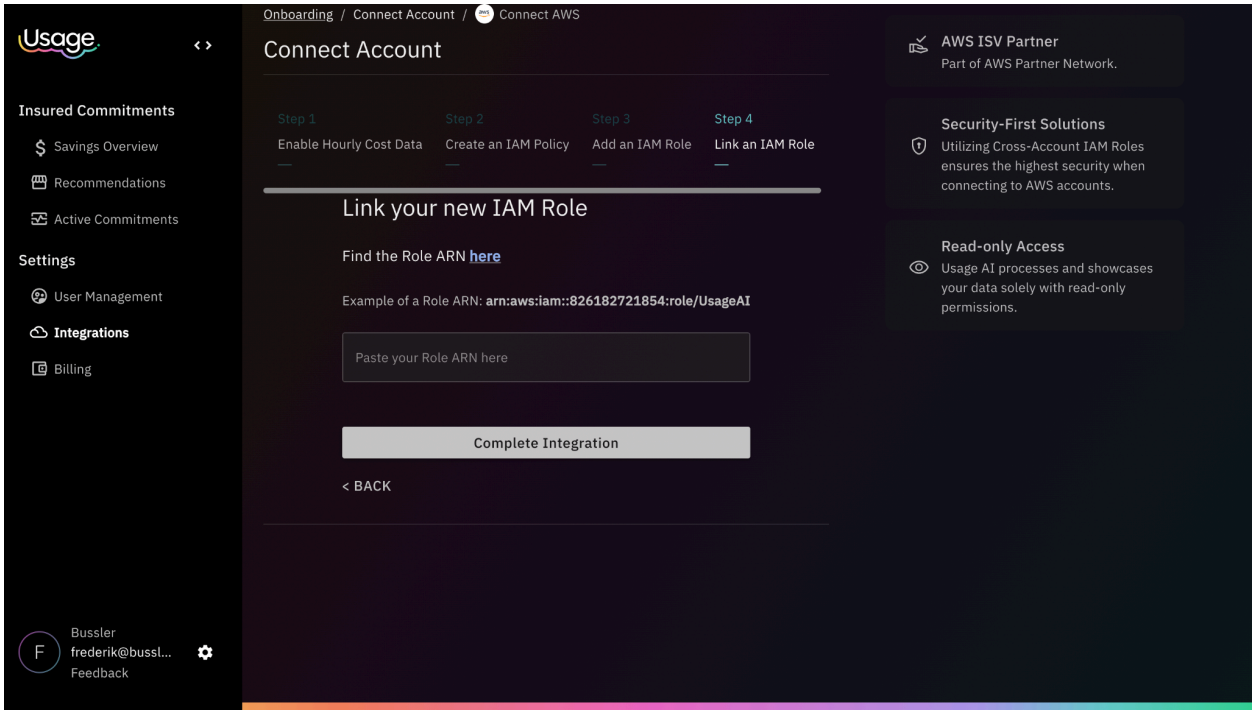


Copy the trust policy from Usage.AI and paste it into the AWS console, replacing the default content. Click Next to move to the permissions screen. Search for "UsageAI" in the policy search box and check the box next to the policy you just created. Click through the remaining screens, name the role "UsageAI" (exactly as shown), and click Create Role.

Step 6: Complete Integration (1 minute)

The role is created and ready to use. In the AWS IAM Roles list, click on the UsageAI role you just created. At the top of the role details page, you'll see the Role ARN—a string that looks like arn:aws:iam::123456789012:role/UsageAI. Copy this entire string.

Return to the Usage.AI dashboard and paste the Role ARN into the field labeled "Paste your Role ARN here." Click Complete Integration. Within seconds, the platform begins analyzing your AWS usage patterns, identifying waste, and calculating savings opportunities.



What a Typical Analysis Reveals

Let me walk you through what you might discover in your savings analysis, using realistic numbers based on hundreds of customer analyses we've performed.

Consider a company spending $200,000 monthly on AWS. This isn't a massive enterprise—it's a successful SaaS company or e-commerce platform with decent traffic and a modern architecture. Here's what their analysis typically reveals:

Their EC2 spend of $94,000 monthly spreads across 40+ instance types, a testament to years of organic growth and experimentation. Different teams launched different services with different instance types, and nobody ever standardized. The analysis shows $31,000 monthly is completely uncovered by any commitments—pure on-demand spending on instances that run 24/7. The platform recommends Compute Savings Plans covering $65,000 of hourly usage, which would save $25,000 monthly with just 27% discounts from one-year terms.

The RDS breakdown tells another story. They're running 14 databases totaling $38,000 monthly. Three are development databases that get torn down and rebuilt regularly—these should stay on-demand. But 11 are production databases that haven't changed size in six months. These stable databases are perfect for Reserved Instances. With 30% discounts from one-year RIs, that's $11,400 in monthly savings on databases alone.

ElastiCache, Redshift, and OpenSearch combine for $46,000 monthly, all running 24/7 with stable configurations that haven't changed in a year. These services are the definition of predictable

workloads. The analysis recommends 80% coverage with one-year RIs, yielding $12,000 in monthly savings with virtually zero risk of overcommitment.

Total potential savings: $48,400 monthly, or $580,800 annually. That's real money that could fund several engineering salaries, accelerate product development, or extend runway. And achieving these savings requires no architecture changes, no service disruptions, and no engineering time beyond the initial 5-minute setup.

The Insurance That Changes Everything

Traditional commitments create an impossible dilemma. Maximum savings require three-year terms, but predicting your infrastructure needs three years out is pure fantasy for growing companies. You might as well predict the weather three years from now—you'll probably be just as accurate.

Usage.AI's Insured Commitments fundamentally change this equation. Instead of you purchasing commitments directly and bearing all the risk, we purchase optimized commitments on your behalf. You immediately get maximum discount rates, often matching three-year commitment discounts. But here's the crucial difference: if your usage drops and those commitments become underutilized, we buy back the unused capacity with actual cash, not credits or vouchers.

This isn't some complex financial instrument or derivative. It's straightforward insurance. You get the savings, we handle the risk. If your company pivots, gets acquired, migrates to Kubernetes, or simply becomes more efficient, you're protected. The cash-back guarantee means you never lose money on unused commitments.

Companies using this approach consistently save 30-50% with zero commitment risk. They get the financial benefits of aggressive commitment strategies without the downside exposure that keeps CFOs awake at night.

Manual vs Automated: Making the Choice

You have three distinct paths for implementing AWS cost optimization, each with different time requirements, risk profiles, and savings potential.

Option 1: Manual Optimization requires significant time investment. Expect to spend 20-40 hours on initial analysis, spreadsheet building, and commitment planning. Then budget 5-10 hours monthly for ongoing management, tracking expiration dates, and adjusting coverage. You own all commitment decisions and their consequences. If you overcommit, the waste is yours. If you undercommit, the missed savings are yours. Teams that execute manual optimization well typically achieve 20-30% savings, but execution quality varies wildly based on available resources and

expertise. This approach works best for companies with dedicated FinOps teams who view cloud cost optimization as a core competency.

Option 2: Usage.AI Free Dashboard provides the analysis and recommendations without the automation. The 5-minute setup gives you complete visibility into your optimization opportunities. You see exactly which Savings Plans and Reserved Instances to purchase, but you execute the purchases yourself in AWS. The platform updates recommendations monthly as your usage evolves, but you need to check it regularly and act on the recommendations. This hybrid approach typically yields 25-35% savings because the recommendations are more sophisticated than manual analysis, but you maintain full control over commitment decisions. It's ideal for teams that want guidance but need to maintain direct control over AWS purchasing decisions for compliance or organizational reasons.

Option 3: Usage.AI Autopilot with Insurance completely automates the optimization process. After the same 5-minute setup, the platform continuously monitors your usage, purchases optimal commitments, manages renewals, and provides cash-back guarantees on any underutilization. You get 30-50% savings with zero risk and zero ongoing time investment. The platform handles everything from initial analysis through commitment lifecycle management. This approach suits teams that want to focus on building products rather than managing spreadsheets, especially high-growth companies where usage patterns change rapidly.

Common Objections Addressed

Every company considering AWS optimization raises similar concerns. Let me address them directly with complete transparency.

1. "We're planning to migrate to Kubernetes soon, so commitments don't make sense for us right now."

This is one of the most common misconceptions. Compute Savings Plans work perfectly with EKS because EKS nodes are just EC2 instances with Kubernetes installed. Your Savings Plan discounts transfer automatically to your containerized workloads. Whether you're running applications directly on EC2, in ECS tasks, EKS pods, or even Lambda functions, Compute Savings Plans apply equally. The only thing that changes is how you deploy applications—the underlying compute still needs optimization.

2. "Our usage is too variable and unpredictable for commitments."

Variable usage is exactly why we recommend 70% coverage with one-year terms instead of 100% coverage with three-year terms. This conservative approach ensures you're only committing to your true baseline—the resources that run consistently regardless of traffic spikes or seasonal variations. The remaining 30% stays on-demand to handle variability. With Insured Commitments, even this conservative approach becomes unnecessary because underutilization gets bought back,

but understanding the principle helps you see why commitment optimization works even for variable workloads.

3. "We already have some Reserved Instances that our previous team purchased."

Existing commitments aren't a problem—they're a head start. Usage.AI's analysis works around your current RIs and Savings Plans, identifying optimization opportunities in your remaining on-demand spend. There's no conflict between old and new commitments. The platform shows you exactly when existing commitments expire and what to replace them with. Many companies find they're sitting on expired commitments they forgot about, paying on-demand rates for resources that were previously covered.

4. "Our finance team needs to approve any commitment purchases, and they're very risk-averse."

The free analysis report includes everything your finance team needs for approval: detailed ROI calculations showing payback periods, risk assessment based on usage volatility, cash flow impact analysis comparing different payment options, and audit trails showing exactly how recommendations were generated. Finance teams actually love the Insured Commitments model because it transforms unpredictable cloud costs into predictable operational expenses with guaranteed savings and zero downside risk.

5. "What if we get acquired or need to shut down services quickly?"

This is where traditional commitments become nightmares and Insured Commitments shine. With standard AWS commitments, you're stuck paying whether you use the resources or not. With Insured Commitments, we buy back unused capacity immediately with cash. Companies going through acquisitions, divestitures, or major pivots get complete protection from stranded commitments. You focus on your business transformation while we handle the financial optimization.

Why Companies Leave Millions on the Table

The mathematics of cloud waste are staggering when you aggregate them across the industry. The average company wastes 35% of their cloud spend through a combination of unused resources, suboptimal pricing, and missed commitment opportunities. For a company spending $100,000 monthly on AWS, that's $35,000 in pure waste—$420,000 annually that could be recovered with proper optimization.

Yet most companies do nothing. They're paralyzed by the complexity of AWS's pricing model, which includes thousands of SKUs across hundreds of services. They're scared of making three-year commitments when they can barely predict next quarter's infrastructure needs.

They're too busy building products and serving customers to become experts in cloud financial management.

The result is a massive transfer of wealth from growing companies to AWS. Every month of inaction is money you'll never recover. While you're evaluating options, running pilots, and seeking approvals, your competitors are reinvesting their cloud savings into product development, marketing, and growth. The opportunity cost compounds over time, creating competitive disadvantages that become harder to overcome.

Start Your Savings Analysis Now

You can know your exact savings potential in the next 5 minutes. No credit card required. No commitment necessary. No sales pressure. Just connect with read-only access and see real numbers based on your actual usage patterns.

The analysis itself often reveals surprising insights beyond just savings opportunities. You'll see which services consume the most resources, which instance types are overprovisioned, and where architectural improvements could yield additional savings. Many companies discover forgotten resources running for months, test environments that became pseudo-production, and instance types that haven't been right-sized since launch.

The companies that have collectively saved $91 million didn't have special advantages or insider knowledge. They just started. They connected their accounts, reviewed the analysis, and made informed decisions about optimization. Whether they chose manual implementation or full automation, they took action instead of accepting the status quo.

Get Your Free Savings Analysis →

What Happens After You Connect

The moment you complete the connection, Usage.AI begins analyzing your entire AWS infrastructure. Within seconds, you see a comprehensive dashboard showing your current spend, identified waste, and savings opportunities across all services. The platform doesn't just show you numbers—it explains them. You understand why certain commitments are recommended, what risks they carry, and how they align with your usage patterns.

The personalized recommendations go beyond generic AWS suggestions. The platform factors in your actual usage volatility, growth trends, and service dependencies. If you have highly variable workloads, it recommends lower coverage levels. If you have rock-solid baseline usage, it suggests more aggressive optimization. Every recommendation includes risk scores and confidence levels so you can make informed decisions.

The risk assessment is particularly valuable for finance teams and executives. You see exactly how much money is at risk with different commitment strategies, what events could trigger underutilization, and how the insurance mechanism protects against losses. This transparency transforms cloud optimization from a mysterious black box into a clear business decision with quantifiable risks and returns.

Implementation options remain flexible even after analysis. You might start with manual implementation to build confidence, then switch to automation once you see the results. Or you might go straight to full automation if you're comfortable with the model. The platform supports whatever approach aligns with your organization's culture and requirements.

Ongoing optimization happens automatically if you choose the autopilot option. The platform continuously monitors your usage, identifies new savings opportunities, and adjusts commitments as needed. You receive monthly reports showing realized savings, upcoming renewals, and optimization recommendations. It becomes a set-and-forget solution that continuously improves your cloud economics without consuming engineering resources.

The Bottom Line

Don’t let confusion, fear of commitments, and your lack of time to optimize cost you big money. A one-year commitment might yield savings of 20-30%, while a three-year commitment offers twice the savings, but that three-year commitment could become a financial disaster if your business evolves.

With Usage.AI, you get maximum savings with zero commitment risk. Our Insured Commitments deliver 30-50% savings while protecting you from any downside through cash-back guarantees. The technology platform handles all the complexity of optimization while the insurance mechanism eliminates financial risk.

The setup takes 5 minutes. The savings last as long as you're on AWS. The risk is completely eliminated through insurance. Every day you wait costs money you'll never recover.

Start Saving in 5 Minutes →