Refine — User Guide

A walkthrough of every page in the Refine application.

Table of Contents

Where to Run Refine

Refine runs anywhere Docker is available. Choose based on your needs:

Option	Best For	Setup
Your laptop/desktop	Evaluation, demos, 1-5 AWS accounts	Extract ZIP, run `setup.bat` (Windows) or `./setup.sh` (Mac/Linux)
AWS EC2 instance	Production use, 5+ accounts, team access	Launch EC2, upload ZIP, run `./setup.sh`

Running locally is the fastest way to get started — under 5 minutes. For always-on monitoring or 5+ accounts, move to an EC2 instance. See DEPLOYMENT_GUIDE.md for full instructions and SYSTEM_REQUIREMENTS.md for sizing.

> ***Local limitation:*** *The nightly data sync only runs when your machine is on. If your laptop sleeps overnight, data won't refresh until you restart Refine. For always-on monitoring, use an EC2 instance.*

Your First Week with Refine

Day 1 — Setup

- [ ] Run the setup wizard (`setup.bat` on Windows, `./setup.sh` on Mac/Linux)
- [ ] Connect your first AWS account (see COMMERCIAL_SETUP_GUIDE.md)
- [ ] Verify the Dashboard loads and shows "Sync scheduled"
- [ ] Check the **AWS Accounts** page — your account should show **Validated**

Days 2–7 — Data Collection

- [ ] The nightly sync runs automatically at 2 AM UTC. Your first full dataset will appear after the first sync completes
- [ ] EC2, RDS, and S3 data should appear within 24 hours
- [ ] Cost data may take 24–48 hours (depends on your S3 bucket population)
- [ ] **Don't worry about empty recommendation pages yet** — recommendations need 14+ days of CloudWatch metrics

Week 2 — Recommendations Start

- [ ] Rightsizing recommendations will begin appearing for EC2 and RDS instances with 14+ days of data
- [ ] Review the **Recommendations** page — each recommendation shows CPU/memory utilization and a confidence score
- [ ] Go to **Savings Config** to set your preferences (commitment term, payment options)
- [ ] Optionally configure the monthly savings email report

Week 4+ — Baselines Lock

- [ ] Resources tracked for 30+ days become "baselined" — savings calculations begin
- [ ] New cost optimization tools are available: Orphaned Resource Scanner, RI Analyzer, S3 Lifecycle Generator, Spot Scorer
- [ ] Review the **Savings Report** for your first monthly summary

Common Questions

Question	Answer
Why are some recommendations grayed out?	Resources under 30 days old are excluded from savings calculations
Why do I see $0 in Cost Overview?	Cost data takes 24–48 hours after first sync. Check that your S3 bucket has data
Why zero RDS recommendations?	RDS needs 14+ days of CloudWatch metrics. Check back after that period
What does "Partial sync" mean?	Some accounts synced but others failed. Check **System Events** for details
How do I check if sync is working?	Go to **AWS Accounts** — the "Last Synced" column shows the timestamp

---

1. Dashboard

URL: `http://localhost:8000/dashboard`

The dashboard is your landing page after login. It gives a high-level overview of your entire AWS environment in one screen.

Sync Status Banner (v2.9.0)

If data collection has failed or data is stale, a warning banner appears at the top of the dashboard showing the sync status and last successful sync time. Click the banner for details or to trigger a manual sync.

Hero Metrics

Four cards across the top:

Metric	What It Shows
Total Monthly Cost	Estimated monthly spend across all connected AWS accounts
Total Resources	Count of all discovered resources across 14 AWS services
Potential Savings	Total monthly savings if all recommendations are applied. Once a savings baseline exists, this card shows **Actual Savings** — the realized cost reduction compared to your baseline month
Optimization Score	0–100 score based on how much waste exists (100 = fully optimized)

Cost Trend

A 12-month bar chart showing your monthly AWS spend over time. Hover over any bar to see the exact dollar amount. The current month is highlighted.

Cost by Service

A horizontal bar chart of your top services ranked by cost. Shows the service name and dollar amount for each.

Resource Inventory

A grid of 14 service tiles. Each tile shows:

- Service icon and name
- Resource count
- Monthly cost (if non-zero)

Click any tile to navigate to that service's detail page.

Top Recommendations

The highest-savings optimization opportunities across all services. Each entry shows the resource name, service type, recommendation action, and estimated monthly savings. Click any recommendation to go to the relevant service page.

Active Savings Plans

Shows your current Savings Plan status. If no plans are active, a link to **Configure Savings** takes you to the Savings Configuration page.

Actions

- **Sync Now** — triggers an immediate data refresh from your AWS accounts
- **Add Account** — opens the AWS account onboarding flow

---

2. Cost Overview

URL: `http://localhost:8000/costs`

A detailed breakdown of your AWS costs by service over the last 7 days.

Summary Cards

Card	Description
Total Cost	Sum of all service costs (updates when you search)
Services Tracked	Number of distinct AWS services with costs
Top Service	The service with the highest cost

Cost Chart

A horizontal bar chart of the top 10 services by cost. Each bar shows the service name and total cost.

Service Cost Table

A searchable, sortable table with columns:

Column	Description
Service	AWS service name
Cost	Total cost in USD
% of Total	Percentage of total spend

Features:

- **Search** — filter by service name (summary cards update live)
- **Sort** — click any column header to sort ascending/descending
- **Refresh Costs** — triggers a fresh data pull from AWS

---

3. Savings Plans

URL: `http://localhost:8000/savings-plans`

Analyzes your current EC2, RDS, Lambda, and Fargate spend and recommends optimal Savings Plan purchases.

Active Savings Plans

The Active Savings Plans table displays all Savings Plans and Reserved Instances you have purchased. Each plan shows:

Column	Description
Type	Compute SP, EC2 Instance SP, Database SP, or RDS RI
Family / Class	Instance family and region (for EC2 Instance SPs) or RDS class (for RDS RIs). Blank for Compute/Database SPs
Hourly	Hourly commitment rate ($/hr)
Monthly	Monthly commitment (hourly x 730 hours)
Payment	No Upfront, Partial Upfront, or All Upfront
Term	1 year or 3 years
Start / End	Commitment period dates
Status	Active (in effect), Queued (future start date), or Expired
Utilization	Percentage of commitment being used by actual compute spend. Green (100%) = fully utilized. Yellow (80-99%) = warning. Red (<80%) = waste

If any plan's utilization is below 100%, a warning banner appears at the top showing the total monthly waste amount.

> ***Tip:*** *Add your active plans after purchasing them so Refine can monitor utilization and warn you before recommendations would cause underutilization.*

Current Status

Four metric cards showing your current Savings Plan utilization, coverage rate, monthly commitment, and realized savings. If you have no active plans, these show 0%.

Spend Analysis

Breaks down your eligible monthly spend by service:

- EC2 Instances
- RDS Databases
- Fargate Tasks
- Lambda Functions
- Total Eligible Spend

Coverage Breakdown

Per-service coverage percentages showing how much of each service's spend would be covered by a Savings Plan.

Scenario Comparison

Side-by-side comparison of two EC2 savings strategies:

Scenario	Description
Compute Savings Plan	Covers EC2, Lambda, and Fargate with moderate discount. Most flexible — applies across instance families and regions
EC2 Instance Savings Plan	Covers EC2 only with higher discount. Committed to specific instance families and regions

Each scenario shows monthly savings, annual savings, hourly commitment, covered services, and discount rate. The active scenario is highlighted. Click a scenario card to switch.

RDS Analysis

Depending on your configured RDS strategy:

- **Database Savings Plan** — shows hourly commitment, discount rate, and projected savings for generation 7+ RDS instances
- **Reserved Instances** — shows a comparison table of 1-year vs 3-year terms across No Upfront, Partial Upfront, and All Upfront payment options

Other Services

Shows Lambda and Fargate spend with a note that these are only covered by Compute Savings Plans.

Commitment Breakdown

A detailed table showing per-instance-family spend and commitment amounts. Toggle between Compute SP and EC2 Instance SP tabs to compare. If EC2 rightsizing recommendations exist, checkboxes let you see how applying them would change your commitment.

Alternative Plans

Plan cards comparing your current configuration against alternative options (different plan types, terms, and payment options). Click any card to apply that configuration.

Purchase Guide

Step-by-step instructions for purchasing. Includes a **Download Purchase Script** button that generates a `.sh` file with the exact AWS CLI commands for your recommended plan. The script includes:

- Savings Plan offering lookup commands
- Purchase commands (commented out for safety — uncomment when ready)
- RDS Reserved Instance or Database Savings Plan commands (if applicable)

> ***Important:*** *Review the purchase script before running it. Purchases are billed immediately and cannot be reversed.*

EC2 Rightsizing Table

Shows EC2 instances that could be resized for cost savings. Columns include current type, recommended type, CPU utilization, current cost, and monthly savings. Checkboxes let you toggle recommendations on/off to see the impact on your Savings Plan commitment.

---

4. Savings Report

URL: `http://localhost:8000/savings-report`

Tracks your actual realized savings over time by comparing your current AWS spend against a historical baseline. This page becomes available after Refine has collected at least two months of cost data.

Hero Cards

Card	Description
Total Saved	Cumulative dollars saved since the baseline date
Baseline Date	The month Refine began tracking savings against
Cost Reduction %	Percentage reduction in monthly spend versus baseline
Months Tracked	Number of months of savings data collected

Cost Trend Chart

A month-by-month line chart comparing your baseline monthly cost against actual spend. The shaded area between the two lines represents realized savings. Hover over any data point to see the exact baseline, actual, and saved amounts for that month.

Savings by Service

A table breaking down total savings by AWS service:

Column	Description
Service	AWS service name
Baseline Cost	Monthly cost at the baseline date
Current Cost	Most recent monthly cost
Monthly Savings	Difference between baseline and current
% Reduction	Percentage improvement

How We Calculate This

Refine establishes a baseline from the first full month of cost data it collects. Each subsequent month is compared against that baseline. Savings shown here reflect actual billing reductions — not projected savings from recommendations.

> ***Note:*** *Savings telemetry requires the* `SAVINGS_ENABLED` *configuration and an S3 bucket for storing cost snapshots. See the setup guide for details.*

Export

- **Download Report (PDF)** — generates a formatted PDF summary of your savings to date, suitable for sharing with stakeholders
- **Export CSV** — downloads the raw month-by-month data for use in spreadsheets or BI tools

---

5. Savings Configuration

URL: `http://localhost:8000/savings-config`

Configure your savings optimization preferences. These settings control how Refine calculates recommendations on the Savings Plans page.

Plan Type

Choose between:

- **Compute Savings Plan** — covers EC2, Lambda, and Fargate (most flexible)
- **EC2 Instance Savings Plan** — covers EC2 only (higher discount)

Commitment Term

- **1 Year** — lower commitment, smaller discount
- **3 Years** — higher commitment, larger discount

Payment Option

- **No Upfront** — pay monthly, smallest discount
- **Partial Upfront** — pay some upfront, medium discount
- **All Upfront** — pay everything upfront, largest discount

Target Utilization

A slider (0–100%) that controls how much of your on-demand spend to cover with a Savings Plan. Default is 80%. Higher means more coverage but higher commitment.

Instance Family Strategy

Three options for EC2 Instance Savings Plans:

Strategy	Description
Single Family	Commit to one instance family (e.g., m6i) — simplest approach
Multi-Family	Select multiple preferred families — Refine optimizes across them
Auto-Optimize	Refine automatically selects the best families based on your workload

When Single or Multi-Family is selected, a categorized family picker appears with tabs: All, General Purpose, Compute Optimized, Memory Optimized, Storage Optimized, Accelerated Computing, High Memory. Each family shows a description and use case.

Force Family Migration

A toggle that tells the optimizer to migrate instances from non-preferred families to your preferred families, even if there is no immediate cost savings. Useful for standardizing your fleet.

RDS Savings Strategy

- **Reserved Instances** — commit to specific instance classes for maximum discount (up to 63% with 3-year all-upfront)
- **Database Savings Plan** — flexible commitment that applies to generation 7+ instances across engines and regions (~20% off)

RDS Instance Classes

Select your preferred RDS instance classes (e.g., db.m6i, db.r6i, db.t3) organized by category.

S3 Data Retention

Number of days to retain S3 data before recommending lifecycle policies. Default: 365 days.

EBS Snapshot Retention

Number of days to retain EBS snapshots before recommending deletion. Default: 90 days.

Savings Breakdown

A real-time summary panel showing projected savings based on your current configuration:

- EC2 Savings
- RDS Savings
- Storage Savings
- Total Monthly Savings

Budget Template

Download a CloudFormation template that creates AWS Budget alerts based on configurable thresholds.

Optimizer Thresholds

The **Optimizer Thresholds** tab lets you tune *how aggressively* Refine flags resources for idle/rightsizing recommendations — without touching code. Each resource class (Compute, Databases, Serverless, Block Storage, Object Storage, Caching, Containers) has its own card with:

- **Preset selector** — *Conservative*, *Default*, or *Aggressive*. Default matches Refine's built-in cutoffs, so changing nothing keeps today's behavior.
- **Advanced overrides** — expand to edit the raw knobs (e.g. Compute's *CPU avg %*, *CPU peak %*, *Memory avg %*, *Network floor*). Editing any knob flips the card to *Custom*.
- **Reset to preset** — discards your overrides and falls back to the canned default.

Thresholds are **shared across clouds per resource class** — the Compute card governs EC2, Azure VMs, and GCE alike. Lower idle cutoffs flag *fewer* resources; raising them (Aggressive) surfaces *more* rightsizing/idle wins. Changes take effect on the next sync.

> *Example: sliding **Compute** to Aggressive raises the idle CPU cutoff, so a VM averaging 8% CPU that was previously "oversized" is now flagged "idle — stop or delete."*

---

6. Recommendations

URL: `http://localhost:8000/recommendations`

A consolidated view of all optimization recommendations across every AWS service.

Summary Cards

Card	Description
Total Monthly Savings	Sum of all recommendation savings (filtered view)
Total Annual Savings	Monthly savings x 12
Total Recommendations	Count of recommendations (filtered view)
High Priority	Count of high-priority recommendations

Recommendations Table

Column	Description
Priority	High (red), Medium (amber), or Low (green) badge
Confidence	High, Medium, or Low badge — based on CPU/memory utilization data availability and consistency (v2.9.0)
Service	AWS service type (EC2, RDS, S3, etc.)
AWS Account	Account ID where the resource lives
Resource	Resource name and ID
Recommendation	What to do and why
Change	Current → Recommended (e.g., `t2.large → t3.medium`)
Monthly Savings	Estimated dollar savings per month

Filters

- **Search** — filter by resource name, ID, description, or account ID
- **Priority** — filter by high, medium, or low
- **Service** — filter by AWS service type
- **Recommendation Type** — filter by action type (rightsize, delete, etc.)

All summary cards update live as you filter.

Actions

- **View CLI** — click on any recommendation row to see the AWS CLI command to implement it. A copy button lets you copy the command to your clipboard.
- **Export Script** — select recommendations with checkboxes and click **Export CLI** to download a `.sh` script with all selected CLI commands, grouped by service. The script covers all 14 supported service types (EC2, RDS, EBS Volumes, EBS Snapshots, S3, NAT Gateways, Load Balancers, Lambda, CloudWatch Logs, Fargate, ElastiCache, Data Transfer, Batch Jobs, and Lightsail). Files are named `refine-recommendations-{accountId}.sh` — one file per AWS account, each containing only that account's recommendations.
- **Exempt** — click the Exempt button on any recommendation to exclude a resource from optimization. You must select a reason (Contractual, Performance, Compliance, Temporary, Shared, Vendor, or Other) and can optionally add notes. Exempted resources move to the Exemptions page and their cost remains in Savings Plan commitment calculations, preserving SP utilization.

> ***Note on idle recommendations:*** *EC2 and RDS idle recommendations are generated even when the instance is already at the smallest available type (t3.micro / db.t4g.micro). In that case, the recommendation is to terminate the instance rather than downsize further.*

Savings Plan Impact

If you have active Savings Plans tracked in Refine, implementing or exempting a recommendation shows a Savings Plan impact notification. This warns you if the action would reduce your SP utilization below 100%, helping you avoid wasted commitment.

7. Exemptions & Resource Policies

URL: `http://localhost:8000/exemptions`

This page has two parts: **Exemptions** (recommendations you've dismissed) and **Resource Policies** (standing per-resource flags). Both keep resources you can't or won't change from cluttering your recommendations — and from skewing your savings totals.

Exemptions are resources you have intentionally excluded from optimization recommendations. Use exemptions when a resource cannot be changed due to contractual, performance, compliance, or other business reasons.

Summary Cards

Card	Description
Total Exemptions	Number of currently exempted resources
Monthly Savings Foregone	Total potential savings you are choosing not to pursue
Temporary	Count of exemptions that will auto-expire and resurface the recommendation

Exempted Resources Table

Column	Description
Resource	Resource name and ID
Service	AWS service type
Recommendation	The recommendation that was exempted
Reason	Why the resource was exempted
Notes	Additional context provided when exempting
Exempted	When the exemption was created
Expires	When the exemption auto-expires (temporary exemptions only), or "Never"
Foregone Savings	Monthly savings this exemption forgoes
Actions	Clear button to remove the exemption and resurface the recommendation

Filters

- **Reason** — filter by exemption reason (Contractual, Performance, Compliance, etc.)
- **Service** — filter by AWS service type

How Exemptions Work

1. **From Recommendations page:** Click the **Exempt** button in the Actions column, select a reason, and optionally add notes

2. **From any service page:** Click the **Exempt** button next to the savings amount for any resource with a recommendation
3. **Temporary exemptions:** Select "Temporary / Planned Change" as the reason to set an auto-expiry (default 90 days). The recommendation reappears after expiry.
4. **Clearing exemptions:** Click **Clear** on the Exemptions page to remove an exemption and resurface the recommendation
5. **Savings Plan impact:** Exempted resources keep their full cost in the SP commitment calculation, ensuring your Savings Plans stay fully utilized

Resource Policies

Below the exemptions table, the **Resource Policies** card lists every standing per-resource flag you've set across AWS, Azure, and GCP. Unlike an exemption (which dismisses one specific recommendation), a policy is a durable rule on the resource itself:

Flag	Effect
Permanent	Suppresses idle / stop / delete recommendations — the resource must stay up (e.g. contractual).
Size-locked	Suppresses downsize / rightsize / tier-change recommendations — the resource needs its exact size.
Exclude from commitments	Drops the resource from Savings Plan / Reserved Instance / CUD sizing, so a short-lived resource doesn't inflate a commitment.

Setting a policy: on any inventory page, use the ⚙ **Assign /** ⚙ **Edit** control in the *Policy* column. Flags are independent — set any combination, optionally with a note. Available on every inventory type (compute, databases, storage, serverless, containers, caching, network, logs) across all three clouds.

What you'll see when a policy blocks a recommendation: the inventory row still shows what the recommendation *would* have saved, but the savings figure is **struck through** with a 📌/🔒 "blocked" marker, and that amount is **excluded** from the page's Potential Savings total and Idle/Oversized count — so locking a resource never inflates your headline numbers.

Removing a policy: click **Remove** on this page, or open the ⚙ control on the inventory row and clear the flags.

---

8. Cost Cleanup

URL: `http://localhost:8000/cost-cleanup`

Find resources by tag and generate CLI cleanup scripts.

How to Use

1. **Select a tag key** from the dropdown (e.g., `environment`, `project`, `team`)
2. **Select one or more tag values** from the checkboxes that appear (e.g., `dev`, `staging`)
3. Click **Find Resources**

Results

Refine queries all 10 supported resource types and shows:

Summary Cards:

- Resource Types found
- Total Resources matched
- Estimated Monthly Savings
- Annual Savings

Resource Groups — one section per resource type with matches:

Resource Type	Type-Specific Columns
EC2 Instances	Instance Type, State
RDS Databases	Engine, Instance Class
EBS Volumes	Volume Type, Size (GB)
EBS Snapshots	Volume Size, Age (days)
NAT Gateways	VPC ID, State
Load Balancers	Type, State
Lambda Functions	Runtime, Memory (MB)
Fargate Tasks	Cluster, vCPU
S3 Buckets	Size (TB), Object Count
CloudWatch Logs	Storage (GB), Retention (days)

Each group has a sortable table with checkboxes. Select the resources you want to clean up.

Download Cleanup Script

Click **Download Cleanup Script** to generate a `.sh` file with AWS CLI commands for all selected resources. The script includes:

- Commands grouped by resource type
- Safety warnings and notes (e.g., "delete database replicas first")
- Tag filter and timestamp in the header

> ***Important:*** *Always review the cleanup script and test in a non-production environment first.*

---

9. AWS Accounts

URL: `http://localhost:8000/accounts`

Manage the AWS accounts connected to Refine.

Accounts Table

Column	Description
Account Name	Friendly name for the account
Account ID	12-digit AWS Account ID
Region	AWS region where the CloudFormation stack is deployed
Status	Validated (green), Pending Setup (yellow), or Error (red)
Primary	Badge indicating the primary account
Last Synced	Timestamp of the last successful data sync

Adding an Account

1. Click **Add Account**
2. Enter your 12-digit AWS Account ID, an optional name, and select a region
3. Click **Add Account**
4. A setup panel appears with:
    - Your **External ID** (copy this for CloudFormation)
    - A **Download CloudFormation Template** button
5. Deploy the CloudFormation template in your AWS account (see the setup guide for details)
6. Once the stack reaches CREATE_COMPLETE, copy the **Role ARN** and **S3 Bucket Name** from the CloudFormation Outputs tab back into Refine
7. Click **Save** — Refine saves your configuration and automatically validates the connection (tests IAM role assumption and S3 bucket access). No separate "Validate" click is needed

Account Actions

- **Details** — view or edit account configuration
- **Set Primary** — mark an account as the primary account
- **Remove** — delete an account (requires confirmation)

Updating an Existing CloudFormation Stack

When Refine releases an update that includes Lambda changes, the AWS Accounts page will alert you automatically — no need to hunt through menus:

- A **yellow banner** appears at the top of the AWS Accounts page listing every account that needs updating
- Each row in the accounts table shows an **"Update CF ⚠"** button in the Actions column and a clickable amber badge in the CF Version column
- The **in-app changelog popup** (shown after a Refine upgrade) also includes a CF update callout with per-account status

To update a stack, click **Update CF ⚠** next to any account (or the CF version badge). This opens a focused update flow:

1. Click **Download Latest Template** to get the current `aws-setup.yaml`
2. In **AWS Console** → **CloudFormation** → select your Refine stack → **Update**
3. Choose **Replace current template** → upload the downloaded template

4. Click through all pages — existing parameters are preserved automatically
5. Acknowledge the IAM capabilities checkbox → **Update stack**
6. Wait for status **UPDATE_COMPLETE**
7. Back in Refine, click **Validate Now** to confirm the update was applied

Updating the stack does **not** delete your S3 bucket or any collected data.

Multiple Accounts

You can connect multiple AWS accounts. Each gets its own CloudFormation stack. Use the AWS account selector in the sidebar to filter data by account across all pages.

---

10. Service Inventory Pages

Each of the 14 AWS services has its own inventory page accessible from the sidebar. All service pages share a common layout:

- **Summary cards** at the top with key metrics
- **Sortable, searchable data table** with service-specific columns
- **Filters** in the sidebar or table header
- **Refresh Inventory** button to trigger a fresh sync
- **Exempt button** on each resource with savings — click to exclude the resource from optimization recommendations (see Exemptions)
- **Empty states (v2.9.0)** — when no data is available for a service, the page shows a helpful message explaining why (e.g., no resources of this type found, sync not yet completed) instead of a blank table

EC2 Instances

URL: `http://localhost:8000/ec2`

Summary Cards: Total Instances, Running Instances, Total Monthly Cost, Total Potential Savings

Summary Table — grouped by instance type showing count, running/stopped split, regions, and costs.

Details Table:

Column	Description
Name	Instance name tag
Account ID	AWS Account ID
Instance Type	e.g., t3.medium, m6i.large
State	Running, Stopped
CPU Avg / Peak	Average and peak CPU utilization across three periods (1-month, 3-month, 1-year) — each period has its own column group
Running Hours	Actual CloudWatch running hours for each period (1mo/3mo/1yr). Requires a CloudFormation stack update for existing accounts — see the AWS Accounts page for instructions
CPU Pattern	Stable, Variable, Spiky, or Idle (based on usage patterns)
Memory %	Memory utilization (if CloudWatch agent is installed)
Network	Network throughput
Monthly Cost	Estimated monthly cost
Rightsizing	Optimal, Oversized, Idle, or Terminate badge
Potential Savings	Estimated savings from applying the recommendation

Filters: Search (name/ID/type), Region, State, Instance Type, Rightsizing Status

Recommendations Panel: A summary of top rightsizing opportunities with estimated savings.

RDS Databases

URL: `http://localhost:8000/rds`

Summary Cards: Total Databases, Available Databases, Total Storage (TB), Total Monthly Cost, Total Potential Savings

Summary Table — grouped by engine (MySQL, PostgreSQL, Aurora, etc.) showing count, Multi-AZ instances, storage, and estimated Reserved Instance savings.

Details Table:

Column	Description
Name	DB instance identifier
Instance Class	e.g., db.m6i.large
Engine	MySQL, PostgreSQL, Aurora, etc.
Status	Available, Stopped
CPU Avg / Peak	CPU utilization (multiple periods)
CPU Pattern	Stable, Variable, Spiky, or Idle
Memory %	Memory utilization
Connections	Average connection count
Monthly Cost	Estimated monthly cost
Rightsizing	Optimal, Oversized, Idle badge
Potential Savings	Estimated savings

Filters: Search (name/ID), Engine, Status, Region, Instance Class

---

S3 Buckets

URL: `http://localhost:8000/s3`

Summary Cards: Total Buckets, Total Data (TB), Total Monthly Cost, Total Potential Savings

Summary Table — grouped by region showing bucket count, total size, object count, and counts of buckets with lifecycle policies and Intelligent-Tiering.

Details Table:

Column	Description
Bucket Name	Clickable link to bucket detail page
Region	AWS region
Size (GB)	Total storage size
Object Count	Number of objects in the bucket
Storage Classes	Badges showing which classes are in use (Standard, Glacier, IA, etc.)
Lifecycle Policy	Yes/No badge
Intelligent-Tiering	Yes/No badge
Monthly Cost	Estimated monthly cost
Recommendation	Add Lifecycle, Enable Intelligent-Tiering, Archive Old Data, or Optimal

Filters: Search (bucket name), Region, Lifecycle Policy, Intelligent-Tiering

Bucket Detail Page (`/s3/{bucket_name}`) — click any bucket name to see storage class breakdown, folder-level analysis, and bucket-specific recommendations.

Recommendations Panel: Highlights buckets without lifecycle policies, without Intelligent-Tiering, large buckets, and archival opportunities.

EBS Volumes

URL: `http://localhost:8000/ebs-volumes`

Summary Cards: Total Volumes, GP2 Eligible for Migration, Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Volume ID	EBS Volume identifier
Volume Type	gp2, gp3, io1, etc. (gp2 highlighted as migration candidate)
Size (GB)	Volume size
IOPS	Provisioned IOPS
State	In-use, Available
Attached Instance	EC2 instance name if attached
Read/Write Ops	Average I/O operations (1-month)
Idle Time %	Percentage of time with no I/O
Monthly Cost	Estimated monthly cost
Recommendation	Keep, Change Type (gp2 → gp3), or Downsize
Potential Savings	Estimated savings

Filters: Search (volume ID/instance name), Region, Volume Type, Recommendation

> ***Tip:*** *gp3 volumes are up to 20% cheaper than gp2 with the same or better performance. Refine identifies all eligible gp2 volumes for migration.*

EBS Snapshots

URL: `http://localhost:8000/ebs-snapshots`

Summary Cards: Total Snapshots, Total Size (TB), Total Monthly Cost, Potential Savings

Details Table:

Column	Description
Snapshot ID	EBS Snapshot identifier
Description	Snapshot description
Volume ID	Source volume
Size (GB)	Snapshot size
Age (Days)	Days since creation (amber if >30d, red if >90d)
State	Completed, Pending
Encrypted	Yes/No badge
Monthly Cost	Estimated monthly cost
Recommendation	Keep, Delete, or Review
Reason	Why the recommendation was made

Filters: Search (snapshot ID/description/volume ID), Region, State, Encrypted, Recommendation

---

NAT Gateways

URL: `http://localhost:8000/nat-gateways`

Summary Cards: Total Gateways, Idle Gateways, Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Gateway ID	NAT Gateway identifier
VPC ID	Associated VPC
Availability Zone	AZ where the gateway is deployed
Data Processed (GB)	Data transferred in the last month
Connection Count	Average connections (1-month)
Idle	Yes/No badge — idle if minimal traffic
VPC Endpoint Opportunities	AWS services that could use VPC endpoints instead of NAT
Orphaned EIP Cost	Cost of unused Elastic IPs associated with this gateway
Monthly Cost	Estimated monthly cost
Recommendation	Keep, Delete, or Add VPC Endpoints
Potential Savings	Estimated savings

Filters: Search (gateway ID/VPC ID), Region, Idle status

> ***Tip:*** *Each idle NAT Gateway costs ~$32/month. VPC endpoints for S3 and DynamoDB are free and eliminate NAT Gateway data processing charges for those services.*

Load Balancers

URL: `http://localhost:8000/load-balancers`

Summary Cards: Total Load Balancers, Idle Load Balancers, Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Name	Load balancer name
Type	ALB (Application) or NLB (Network) badge
Scheme	Internet-facing or Internal
State	Active, Provisioning
Request Count	Total requests in the last month
Healthy / Unhealthy Targets	Target group health status
Target Group Count	Number of associated target groups
Idle	Yes/No badge
Monthly Cost	Estimated monthly cost
Recommendation	Keep, Delete, or Consolidate
Potential Savings	Estimated savings

Filters: Search (LB name), Region, Type, Idle status

Lambda Functions

URL: `http://localhost:8000/lambda`

Summary Cards: Total Functions, Overprovisioned Count, Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Function Name	Lambda function name
Runtime	Node.js, Python, Java, etc.
Memory (MB)	Allocated memory
Timeout (s)	Function timeout
Provisioned Concurrency	Count or "None"
Avg / Max Duration	Execution time in milliseconds
Max Memory Used	Peak memory consumption
Memory Utilization %	How much of allocated memory is used (color-coded)
Invocations	Total invocations (1-month)
Errors / Throttles	Error and throttle counts
Monthly Cost	Estimated monthly cost
Recommendation	Optimal, Downsize Memory, or Remove Provisioned Concurrency
Potential Savings	Estimated savings

Filters: Search (function name), Region, Runtime, Recommendation

---

CloudWatch Logs

URL: `http://localhost:8000/cloudwatch-logs`

Summary Cards: Total Log Groups, Total Storage (GB), Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Log Group Name	CloudWatch Log Group name
Retention	Current retention period (color-coded: Never=red, ≤30d=green, 30-90d=amber, >90d=red)
Storage (GB)	Amount of log data stored
Write Frequency	Average writes per hour (1-month)
Last Event	Date of last log event
Unused	Yes/No badge — no recent events
Recommended Retention	Suggested retention period
Monthly Cost	Estimated monthly cost
Recommendation	Keep, Shorten Retention, or Delete
Potential Savings	Estimated savings

Filters: Search (log group name), Region, Unused status, Recommendation

> ***Tip:*** *Log groups set to "Never Expire" accumulate storage costs indefinitely. Setting a retention policy is one of the easiest cost wins.*

Fargate Tasks

URL: `http://localhost:8000/fargate`

Summary Cards: Total Tasks, Spot Eligible, Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Task Family	Task definition family name
Cluster / Service	ECS cluster and service name
vCPU / Memory	Allocated compute resources
CPU Utilization	Average and peak CPU (color-coded)
Memory Utilization	Average and peak memory (color-coded)
Uptime	Average hours per day
Restart Count	Restarts in the last month
Spot Eligible	Yes/No badge
Monthly Cost	Estimated monthly cost
Recommendation	Optimal, Downsize, or Migrate to Spot
Potential Savings	Estimated savings

Filters: Search (task family/service), Region, Cluster, Recommendation

---

ElastiCache

URL: `http://localhost:8000/elasticache`

Summary Cards: Total Clusters, Idle/Oversized Count, Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Cluster ID	ElastiCache cluster identifier
Engine	Redis or Memcached
Node Type	e.g., cache.m6g.large
Num Nodes	Number of nodes in the cluster
Multi-AZ	Yes/No badge
CPU Avg / Peak	CPU utilization
Cache Hit Ratio	Percentage of requests served from cache (>90% good, <70% low)
Evictions/sec	Average evictions per second
Connections	Average connection count
Idle	Yes/No badge
Monthly Cost	Estimated monthly cost
Recommendation	Optimal, Downsize, Upsize, or Delete
Potential Savings	Estimated savings

Filters: Search (cluster ID), Region, Engine, Recommendation

Data Transfer

URL: `http://localhost:8000/data-transfer`

Summary Cards: Total Flows, Total Data Transferred (GB), Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Traffic Type	Cross-Region, Cross-AZ, NAT→S3, NAT→DynamoDB, or Internet Egress
Source / Dest Region	Origin and destination regions
Data Transferred (GB)	Volume of data in the last month
Monthly Cost	Estimated transfer cost
VPC Endpoint Available	Yes/No — whether a VPC endpoint could replace this flow
Recommended Action	Add VPC Endpoint, Consolidate Regions, or No Action
Potential Savings	Estimated savings

Filters: Search (regions/service), Region, Traffic Type, Recommended Action

> ***Tip:*** *VPC endpoints for S3 and DynamoDB are free and eliminate NAT Gateway data processing charges. This is often one of the largest savings opportunities.*

Batch Jobs

URL: `http://localhost:8000/batch-jobs`

Summary Cards: Total Jobs, Spot Eligible, Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Job Name	Batch job definition name
Queue / Compute Env	Job queue and compute environment
vCPU / Memory	Allocated resources
Memory Utilization %	How much memory is actually used (color-coded)
CPU Utilization %	How much CPU is actually used (color-coded)
Job Count	Number of runs in the last month
Avg Runtime	Average execution time
Failed Count	Number of failures
Spot Eligible	Yes/No badge
Monthly Cost	Estimated monthly cost
Recommendation	Optimal, Migrate to Spot, or Downsize Memory
Potential Savings	Estimated savings

Filters: Search (job name/compute environment), Region, Recommendation

---

Lightsail

URL: `http://localhost:8000/lightsail`

Summary Cards: Total Instances, Running Instances, Total Monthly Cost, Total Potential Savings

Details Table:

Column	Description
Instance Name	Lightsail instance name
Bundle ID	Current bundle (e.g., medium_3_0)
Blueprint	OS/application blueprint
State	Running, Stopped
vCPUs / RAM / SSD	Instance resources
CPU Avg / Peak	CPU utilization (color-coded)
Network	Average throughput
Last Active	Date of last activity
Monthly Cost	Bundle cost
Recommendation	Optimal, Downsize, Idle—Stop, or Delete
Potential Savings	Estimated savings

Filters: Search (instance name/bundle), Region, State, Recommendation

---

11. User Management

URL: `http://localhost:8000/user-management`

This page is available only to users with the **ROOT** or **ADMIN** role. It provides two tabs — **Users** and **Groups** — for managing who can access your Refine server and what accounts they can see.

User Roles

Refine supports three roles:

Role	Description
ROOT	Server administrator. Can manage all users and groups, sees all AWS accounts, and can take any action across all accounts. One per server — the person who runs the setup wizard becomes the ROOT user.
ADMIN	Group administrator. Assigned to groups by Root. Can manage users within their groups, create other admins scoped to their own groups, and take full actions within group accounts.
USER	Standard user. Assigned to groups by Root or Admin. Full actions within group accounts (exempt, modify recommendations, etc.). Cannot manage users or groups.

Users Tab

Create and manage user accounts.

Creating a user:

1. Click **Add User**
2. Enter the user's email address and a password
3. Select a role (ROOT, ADMIN, or USER)
4. Assign the user to one or more groups (determines which AWS accounts they can see)
5. Click **Create User**

Editing a user: Click any user row to change their role, password, or group assignments.

Removing a user: Click **Remove** in the Actions column (requires confirmation).

Groups Tab

Groups are named collections of AWS accounts. Users inherit access to all accounts in their assigned groups.

Creating a group:

1. Click **Add Group**
2. Enter a group name (e.g., "Production Team", "Dev Team", "Finance")
3. Select the AWS accounts that belong to this group
4. Click **Create Group**

Editing a group: Click any group row to add/remove accounts or rename the group.

Removing a group: Click **Remove** in the Actions column. Users previously assigned to the group lose access to those accounts unless they belong to another group that includes them.

How Access Works

- The **ROOT** user sees all AWS accounts regardless of group assignment.
- An **ADMIN** or **USER** sees only the accounts included in the groups they belong to. If a user belongs to multiple groups, they see the union of all accounts across those groups.
- Admins can create other admins, but only assign them to groups they themselves own.
- All users on a Refine server share the same underlying data — groups control visibility, not data collection.

Directory Integration (LDAP / Azure AD)

Refine can integrate with your organization's Active Directory or Azure AD so employees sign in with their corporate credentials instead of separate passwords.

- **On-Prem AD (LDAP):** Employees enter their AD username and password on the Refine login page. Refine authenticates against your AD server directly.
- **Azure AD / Entra ID:** A "Sign in with Microsoft" button appears on the login page. Employees are redirected to Microsoft for authentication and returned to Refine.

Directory users are **auto-provisioned** on first login — no manual account creation needed. AD group memberships can be mapped to Refine account groups, so team members automatically see the right AWS accounts.

The ROOT user always uses local email/password authentication regardless of directory configuration.

To configure: Go to **Directory Settings** in the sidebar (ROOT only). See LDAP Setup Guide or Azure AD Setup Guide for step-by-step instructions.

---

12. System Events

URL: `http://localhost:8000/system-events`

A persistent log of system events — sync failures, S3 read errors, IAM issues, and other operational problems. Each event includes an error code linked to a knowledge base entry explaining the likely cause and recommended fix.

Event List

Column	Description
Severity	Critical (red), Error (amber), Warning (yellow), or Info (blue) badge
Error Code	Machine-readable code (e.g., `SYNC_FAILED`, `S3_READ_ERROR`) — click to see the knowledge base entry
Message	Human-readable description of what happened
Source	The service or component that raised the event
Occurrences	How many times this event has been seen (deduplicated)
First Seen	When the event was first recorded
Last Seen	Most recent occurrence
Status	Active, Acknowledged, or Resolved

Actions

- **Acknowledge** — mark an event as seen without resolving it
- **Resolve** — manually close an event
- **Export for Support** — download all events as a JSON file to share with Blacktip Solutions. The export contains only event data (error codes, messages, timestamps) — no credentials or sensitive configuration are included.

> *Events auto-resolve when the underlying issue is fixed (e.g., a sync that previously failed completes successfully). Repeated occurrences of the same error increment the occurrence count rather than creating separate entries.*

---

13. Common Features

These features are available on every page in Refine.

Sidebar Navigation

The left sidebar provides quick access to all pages:

- **Dashboard** — overview of your entire AWS environment
- **Cost & Savings** — Cost Overview, Savings Plans, Savings Report, Savings Config, Recommendations, Exemptions, Cost Cleanup
- **System** — System Events, AWS Accounts, User Management (admin only)

- **14 Service Pages** — one for each AWS service Refine monitors (EC2, RDS, S3, EBS Volumes, EBS Snapshots, NAT Gateways, Load Balancers, Lambda, CloudWatch Logs, Fargate, ElastiCache, Data Transfer, Batch Jobs, Lightsail)

Click the collapse arrow at the bottom to minimize the sidebar.

AWS Account Selector

If you have multiple AWS accounts connected, use the account selector in the top-right to filter data by account. Select **Deselect All** to see data from all accounts, or check specific accounts to focus on them.

Tag Filtering

Available on inventory pages. Use the tag filter sidebar (bottom-right) to filter resources by tag key and value:

1. Select a tag key from the dropdown
2. Check one or more tag values
3. Click **Add Filter**

Filters use AND logic across keys and OR logic within values. Active filters appear as chips that you can remove individually. Filters persist across page loads.

Sync Now

Click **Sync Now** in the top bar to trigger an immediate data refresh from your AWS accounts. The Lambda function in your AWS account collects data from 14 services in parallel across all regions. First syncs may take a few minutes.

Version and License

The bottom-left corner of the sidebar shows:

- **Refine version** (e.g., Refine v1.2.4)
- **License status** — color-coded badge showing your license expiry date

Badge Color	Meaning
Green	License valid
Amber	Expiring soon (within 30 days)
Red	License expired
Purple	Demo mode

Need Help?

Contact Blacktip Solutions at **support@blacktipsolutions.com**