

AWS MODERNIZATION FOR STARTUP AND GROWTH TEAMS

Startup AI & Data Modernization

Build agentic AI and modernize your startup stack on AWS, with governance and a measured path to lower operating costs.

AI ECONOMICS

Spend on useful outputs

Move suitable OpenAI workloads to Amazon Bedrock, right-size models, and measure cost at the quality your product needs.

DATA FOUNDATIONS

Keep the data working

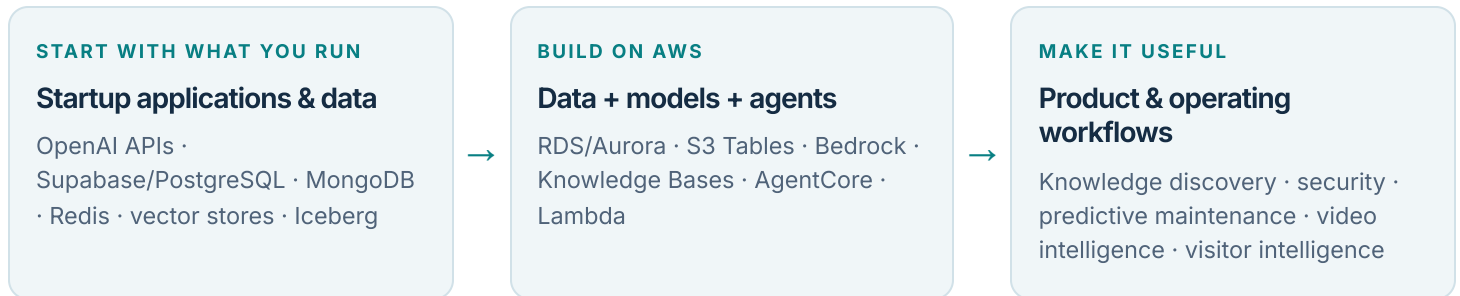
Map Supabase, hosted PostgreSQL, Iceberg, and other startup services to the AWS architecture that fits the workload.

AGENTIC PROGRAMS

Deliver with governance

Build and operate agents with Bedrock AgentCore, defined permissions, human review, and evidence your team can inspect.

One delivery path, from today's stack to production



Across the stack: identity and permissions · encryption · observability · cost attribution · tested cutover and rollback

CUSTOMER EXPERIENCE



AWS PERSPECTIVE AWS names Tactical Edge among partners offering alternatives to Amazon Monitron. [↗](#)

Start a conversation.

sales@tacticaledgeai.com →

PRODUCT INTELLIGENCE. MARKETING. SECURITY.

Customer experience. AWS execution.

Real engagement focus, followed by AWS patterns that can extend the work.

Squid · asksquid.ai



Visitor intelligence · AWS-native AI engine

Tactical Edge helped build the AWS-native generative AI engine that indexes Squid's platform data, the backbone behind its visitor-identification and intelligence product.

AWS extension. Bedrock and AgentCore can build on that indexed context to explain observed activity and prepare tenant-scoped CRM tasks for review.

Value: Make platform data useful in the product and its customer workflows.

Clipr



Video intelligence · metadata on AWS

Tactical Edge created video metadata on AWS using Bedrock, alongside transcription and timeline-alignment workflows for searchable media.

AWS extension. Connect S3, Transcribe, Translate, Bedrock enrichment, and timestamped search; AgentCore connects the workflow's approved tools.

Value: Turn recordings into searchable, reusable information.

Arnica



Security · contextual remediation

Tactical Edge connected security findings, application context, and reviewed remediation workflows around the existing security product.

AWS extension. Bedrock explains findings; Knowledge Bases retrieves approved guidance; AgentCore connects issue and repository tools with scoped access.

Value: Give developers actionable, traceable security context.

Broughton Partners



Marketing · campaign and intake operations

Tactical Edge supported security and cloud cost practices for Broughton Partners, a legal-marketing and intake business.

AWS extension. Use Bedrock to summarize approved campaign and intake data. AgentCore prepares CRM tasks; people review outreach and publishing.

Value: Connect marketing insight to reviewed action.

AWS extensions describe possible next steps, with service selection and delivery scope agreed for each customer. They are distinct from the engagement work summarized above.

Agentic programs with governance

Agent Readiness Assessment · Agent Build Program · Managed Agent Operations

Identity, data access, tool controls, human review, and operating evidence travel with the workflow.

Bring one product or operating challenge.

Start a conversation: sales@tacticaledgeai.com

KNOWLEDGE. MAINTENANCE. CONNECTIVITY.

Practical AI across operating environments.

Use the organization's information, people, and controls as the starting point.

Tyler Union



Predictive maintenance · Amazon Monitron

Tactical Edge implemented Amazon Monitron to bring equipment-condition monitoring into the manufacturer's maintenance workflow.

AWS extension. New monitoring designs can combine industrial sensors, AWS IoT Core, and SageMaker AI; Bedrock explains alerts and AgentCore prepares maintenance tasks.

Value: Turn condition signals into reviewed maintenance action.

EMK Institute

EDWARD M. KENNEDY INSTITUTE
FOR THE UNITED STATES SENATE

Knowledge management · AI

Tactical Edge helped the Institute apply AI to institutional knowledge and staff information workflows.

AWS extension. Bedrock and Knowledge Bases support source-grounded answers; AgentCore connects approved tools; Amazon Quick provides a staff workspace.

Value: Make trusted institutional information easier to find and use.

US Tronics



Database workflows · agent harness

Tactical Edge helped US Tronics shape database workflows and an agent harness around its communications business.

AWS extension. Bedrock interprets tasks; AgentCore connects approved database tools. Scope queries, validate results, and review changes before execution.

Value: Make database work useful, traceable, and controlled.

THE NEXT USE CASE

Build around the way your team works.

Connect approved knowledge and tools to one useful task, with clear limits and human oversight.

- A named workflow and operating owner
- Approved data and application access
- A baseline and a representative pilot
- A release and governance plan

sales@tacticaledgeai.com →

Tyler Union's delivered implementation used Monitron. Monitron closed to new customers in 2024; new projects use a separately assessed sensor, AWS IoT, and predictive-model design.

OPENAI WORKLOADS → AMAZON BEDROCK + AGENTCORE

Make the AWS business case measurable.

Compare the same product workload, output quality, latency, and reliability. Account for models, retrieval, agent execution, data transfer, operations, and migration.

01 Baseline the real workload

Inventory models, prompts, tokens, tools, vector stores, concurrency, error rates, and monthly usage and cost. Define cost per accepted answer, action, or processed video hour.

02 Choose the right migration path

Use Bedrock's OpenAI-compatible APIs where the selected model and features fit, or adapt the application to Bedrock-native APIs. Rework state, tools, embeddings, and evaluation as needed.

03 Prove quality and savings

Replay representative tasks. Compare groundedness, task completion, p95 latency, failure recovery, and all-in cost before routing production traffic.

04 Operate the product on AWS

Use AgentCore for agents and permitted tools, CloudWatch for traces, and budgets and cost allocation for ongoing economics.

ILLUSTRATIVE ECONOMICS · ASSUMED INPUTS

From a cost target to a decision

Planning example only. These are neither customer results nor AWS price quotes.

Monthly recurring cost	Assumption
Current complete stack	\$12,000
AWS model inference	\$5,000
AWS retrieval, storage, network	\$1,000
Agent and workflow execution	\$1,000
Operations, security, support	\$1,500
AWS monthly total	\$8,500

Assumed one-time migration and validation: **\$14,000**.
 Replace every input with your baseline, AWS estimates, and pilot measurements.

29.2%

Illustrative recurring reduction
 \$3,500 per month

4 months

Illustrative migration payback
 $\$14,000 \div \$3,500$

\$28,000

Illustrative first-year net savings
 $12 \times \$3,500 - \$14,000$

Right-size the model

Benchmark Amazon Nova and other suitable Bedrock models against the task. Use supported routing where it fits.

Choose an inference mode

Test prompt caching for repeated context and batch processing for asynchronous work where supported.

Consolidate selectively

Evaluate existing AWS data stores for retrieval and simplify integrations when the full operating cost improves.

Savings = current total recurring cost – AWS total recurring cost. Payback = one-time migration cost ÷ monthly savings, when positive. Model, API, feature, and Region availability are assessed for each workload.

COMMON STARTING POINTS. EVALUATED AWS DESTINATIONS.

Bring the startup stack into AWS.

Validate application compatibility and workload economics before choosing your AWS target.

Current stack	AWS destination to evaluate	Migration decision and potential cost lever
OpenAI APIs and custom AI orchestration	Amazon Bedrock + Bedrock AgentCore	Validate model outputs, tool calls, streaming, state, quotas, and supported API features. Model selection, caching, batch work, fewer bespoke integrations.
Supabase, Neon, or other hosted PostgreSQL	Amazon RDS for PostgreSQL or Aurora PostgreSQL	Check extensions, roles, RLS, connections, and application services separately. Right-sized capacity, pooling, selective platform consolidation.
Apache Iceberg and external lakehouse catalogs	Amazon S3 Tables, or S3 + AWS Glue Data Catalog	Validate catalogs, table versions, permissions, writer jobs, and query behavior. Managed maintenance, storage optimization, efficient scans.
MongoDB / MongoDB Atlas	Amazon DocumentDB, or retain Atlas on AWS where appropriate	Test drivers, indexes, aggregation, transactions, and functional differences. Compare the real workload and feature costs before moving.
Redis, Valkey, or Upstash-backed caches	Amazon ElastiCache for Valkey or Redis OSS	Validate commands, TTL, eviction, TLS clients, clustering, and reconnection. Engine choice, capacity fit, connection and cache tuning.
Pinecone, Qdrant, Weaviate, or custom vector stores	Aurora PostgreSQL with pgvector, or Amazon OpenSearch	Benchmark retrieval quality, filters, update rates, tenancy, latency, and scale. Reuse suitable data infrastructure or select a dedicated index.
Snowflake / BigQuery analytics workloads	Amazon Redshift or Amazon Athena on S3	Port SQL and pipelines; reconcile business totals; replay representative queries. Workload sizing, columnar scans, partition and query tuning.
Heroku, Render, Vercel, or custom app hosting	AWS Amplify, Lambda, or Amazon ECS on Fargate	Check framework behavior, edge/runtime assumptions, jobs, networking, and releases. Match execution and scaling to the application's demand.

Move the application contract with the data. Include authentication, APIs, event delivery, secrets, backups, recovery, observability, and the cost of any services being replaced.

OPEN FOUNDATIONS. AWS OPERATIONS.

Keep PostgreSQL. Keep Iceberg. Change the operating model.

Move the platform around the data deliberately, then measure the cost and operational effect.

Supabase → PostgreSQL on AWS

Supabase already uses PostgreSQL. The move is to an AWS database and operating model, with separate plans for the surrounding backend services.

Database	RDS or Aurora PostgreSQL. Validate extensions, SQL, indexes, roles, pooling, and row-level security.
Auth	Evaluate Cognito or retain the identity provider. Map JWT claims and re-test authorization policies.
Storage	Move objects to S3 and validate access controls, signed URLs, and application behavior.
Realtime	Evaluate AppSync or WebSocket/event services. Rebuild required presence, broadcast, and change semantics.
Edge + APIs	Evaluate Lambda/API Gateway or ECS. Preserve API contracts or explicitly refactor clients and runtime dependencies.

Cost case: compare the complete Supabase plan with AWS database, network, identity, storage, application runtime, and operations costs. Use an engine-native transfer or validated replication path that fits the downtime window.

Apache Iceberg → AWS lakehouse

Retain the open table format. Choose the table storage, catalog, maintenance, and analytics approach that fits your writers and query engines.

Existing Iceberg tables + catalog + jobs

Inventory data paths, snapshots, versions, and permissions.

MANAGED TABLE PATH

Amazon S3 Tables

Iceberg tables with managed compaction and snapshot maintenance. Integrate the catalog and validate engine compatibility.

GENERAL-PURPOSE PATH

S3 + Glue Data Catalog

Retain more control over the existing layout and table lifecycle. Configure supported table optimization and governance.

Query and process with Athena, EMR, or compatible analytics engines.

Cost case: measure storage, requests, compaction, retention, and compute or scanned bytes. Validate schema evolution, deletes, concurrent writers, representative queries, and reconciliation before switching jobs.

The migration proof covers data and behavior

Inventory & decide

Dependency map, extension and catalog checks, AWS target and TCO options.

Replay & reconcile

Representative queries, application tests, business totals, permissions, and recovery.

Rehearse & cut over

Migration runbook, downtime or replication plan, named decision, rollback, and stabilization.

FROM A USEFUL TASK TO AN OPERATED SYSTEM

Agentic programs. Governance built in.

Start with readiness, deliver a bounded workflow, and operate it with clear ownership. Choose Amazon Bedrock, Bedrock Agents, AgentCore, and Amazon Quick according to the work and the environment.

DISCOVER

Agent Readiness Assessment

Prioritize use cases, connect business value to data and tools, and define the operating and governance requirements.

Deliverables: readiness findings, a scoped pilot, success measures, and a delivery roadmap.

BUILD

Agent Build Program

Design and integrate an agent workflow, evaluate it against representative tasks, and rehearse a controlled rollout.

Deliverables: a tested workflow, permission model, human-review points, release plan, and handoff.

OPERATE

Managed Agent Operations

Watch quality, latency, cost, tool behavior, and exceptions. Evaluate model and workflow changes before release.

Deliverables: monitoring, incident ownership, operating records, and an improvement backlog.

Controls that travel with the workflow

Identity + data boundaries

Scope identities, tenant access, approved sources, secrets, encryption, and retention.

Tools + human oversight

Limit actions, validate tool inputs, and route consequential changes to a named reviewer.

Evaluation + release

Test retrieval, output quality, tool use, failure paths, and regressions before expanding access.

Decision + action records

Retain useful evidence of tool calls, reviews, overrides, and exceptions with sensitive-data controls.

Monitoring + recovery

Define alert thresholds, escalation, stop behavior, rollback, and a responsible operating owner.

Cost + business value

Measure useful outputs and workflow outcomes alongside model, agent, retrieval, and operating cost.

AWS building blocks: IAM · KMS · Secrets Manager · CloudTrail · CloudWatch · Bedrock Guardrails · AgentCore identity and observability · application-level policy and approvals

[Agent Programs](#) → [Agent Governance](#) →

ASSESS → PROVE → DELIVER

A practical offering you can start with.

Begin with one business-critical workload. Use the evidence from that pilot to size the migration and extend the AWS platform.

PACKAGE 01

Stack & AI economics assessment

- Application and data inventory
- Model and API workload baseline
- AWS target and compatibility map
- TCO, risk, and migration options

You receive: a decision-ready roadmap, cost model, and scoped pilot.

PACKAGE 02

Representative workload pilot

- One AI or data migration path
- Quality and business-parity tests
- Measured latency and unit cost
- Identity and tool integration

You receive: a working pilot, test evidence, and production recommendation.

PACKAGE 03

Migration & production operations

- Migration waves and cutover
- Rollback and recovery rehearsal
- Monitoring and cost attribution
- Stabilization and team handover

You receive: an operated AWS workload with runbooks and ownership.

The AWS foundation behind the engagement

DATA + EVENTS

Bring product context together

S3 · RDS/Aurora · S3 Tables/Glue · application APIs · IoT or event streams



MODELS + RETRIEVAL

Make information usable

Bedrock + Knowledge Bases; Transcribe/Translate for media; SageMaker AI for predictive models



AGENTS + EXECUTION

Connect approved actions

AgentCore Runtime, Gateway, Identity, Observability; Lambda/Step Functions; application tools

IAM + KMS + Secrets Manager · permission-aware retrieval · Guardrails for model content · application approvals for consequential actions

Start a conversation.

Tell us what you want to improve, the workflow involved, and the systems your team uses.

sales@tacticaledgeai.com

tacticaledgeai.com →

Service and model selection follows the workload's Region, data boundary, compatibility, and economics. Scope, timing, and commercial terms are agreed for each engagement.