Flink Training for Real-Time Data Engineering

Deploying and Operating Flink on AWS

AGENDA

- Comparing deployment models: Self-hosted vs. Managed Services.
- **Focus 1: Amazon Managed Service for Apache Flink**
  - Setup, configuration, and security.
  - Integration with other AWS services (S3, IAM, CloudWatch).
  - Autoscaling and monitoring a managed Flink application.
- **Focus 2: Running Flink on Kubernetes (Amazon EKS).**
  - Overview of the Flink Kubernetes Operator.
  - Best practices for configuring Flink clusters on EKS.
  - Managing Flink job lifecycles (deployment, updates, scaling) via Kubernetes.

Why Apache Flink on AWS?

Industry-Leading Performance

Stream processing with exactly-once semantics and ultra-low latency capabilities that set the industry standard.

Proven at Scale

Powers real-time analytics at global leaders like Uber, Netflix, and Goldman Sachs with mission-critical workloads.

AWS Flexibility

Offers both self-hosted and managed deployment options, tailored to different organisational needs and use cases.

Self-hosted Flink on AWS: What It Looks Like

Deploy on Your Infrastructure

Deploy Flink clusters on EC2 instances or Kubernetes (EKS) with complete architectural control over your streaming infrastructure.

Full Configuration Control

Manage JobManager, TaskManagers, and cluster configuration with granular customisation for your specific requirements.

Manual Operations Required

Handle setup of high availability, scaling policies, monitoring systems, and version upgrades through your DevOps processes.

Automation Tools Available

Leverage tools like Flink-deploy or Ansible scripts to provision and manage EC2 clusters systematically.

Managed Service for Apache Flink on AWS

Fully Managed Environment

Serverless Flink with autoscaling and durable state management handled automatically by AWS.

Integrated Development

AWS Studio notebooks provide interactive development environment for rapid prototyping and testing.

Seamless AWS Integration

Built-in connectivity with Kinesis, MSK, S3, and automatic checkpointing with fault tolerance.

Accelerated Production

Eliminates cluster management overhead, dramatically reducing time from development to production deployment.

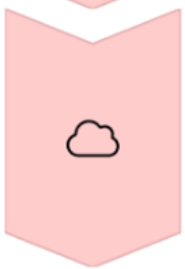
Deployment Modes: Side-by-Side Comparison

Feature	Self-hosted (EC2/EKS)	Managed Service (AWS)
Cluster Management	Manual setup and maintenance	Fully automated by AWS
Scaling	Manual or custom automation	Autoscaling based on throughput
Fault Tolerance	Requires ZooKeeper or Kubernetes HA	Built-in durable state & snapshots
Development Workflow	IDE + CLI + manual deployment	Studio notebooks + one-click deploy
Cost Model	Pay for EC2/EKS resources	Pay per usage, no idle cluster cost

Operational Complexity & Best Practices

Self-hosted Requirements

Demands expertise in cluster provisioning, networking, security groups, IAM roles, and comprehensive monitoring with CloudWatch or Prometheus.



Managed Simplification

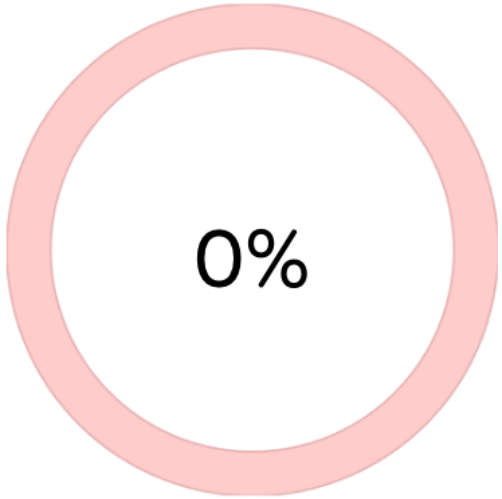
Streamlines operations with predefined IAM roles, seamless VPC integration, and built-in logging and metrics collection.

Managed Service requires VPC internet access for durable state functionality, though alternatives exist for private VPC configurations. Both deployment models require careful planning for source and sink integration with services like Kinesis, MSK, S3, and Glue Catalog.

Cost Considerations

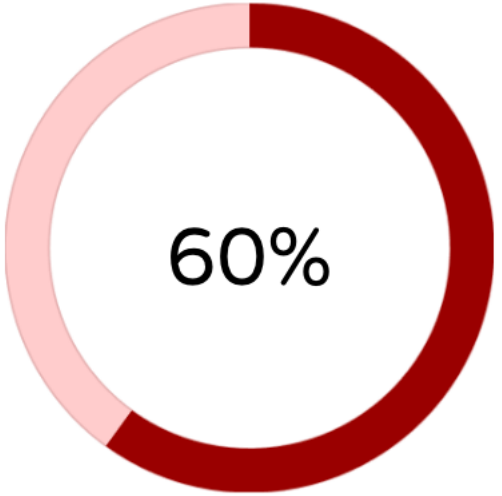
Self-hosted Costs

Fixed EC2/EKS costs regardless of workload utilisation, with potential for overprovisioning.



Managed Idle Costs

Pay-as-you-go with no upfront cluster costs; expenses scale directly with data volume and parallelism.



Operational Savings

Managed Service reduces operational overhead and eliminates risks from misconfiguration.

DATACOUCH
CREATING INFINITE POSSIBILITIES

DSAI
Data Science and AI Academy

Piramal Finance

Focus 1:

Amazon Managed Service for Apache Flink

What is Amazon Managed Service for Apache Flink?

Fully Managed Service

Build and run real-time stream processing applications without managing infrastructure complexity.

Multi-Language Support

Supports Apache Flink APIs in Java, Scala, Python, and SQL for flexible development approaches.

Enterprise Features

Provides exactly-once processing, stateful computations, and high availability with multi-AZ replication.

Security Fundamentals: Shared Responsibility Model

Infrastructure Security

AWS secures the cloud infrastructure whilst you secure your applications and data in the cloud.

Encryption Standards

All API calls use TLS 1.2+ with perfect forward secrecy (ECDHE/DHE) for maximum protection.

Access Control

Use IAM roles for fine-grained access control and temporary credentials—avoid embedding long-term keys.

Compliance Monitoring

Enable server-side encryption on dependent resources and monitor API activity with AWS CloudTrail.

Autoscaling Your Flink Application: Why It Matters

Dynamic Workloads

Streaming workloads fluctuate; autoscaling ensures performance and cost efficiency throughout varying demand cycles.

Lag Monitoring

Monitor key metrics like records_lag_max and millisBehindLatest to detect processing delays before they impact performance.

Resource Optimisation

Use CPU Utilization and heap Memory Utilization metrics to anticipate resource bottlenecks proactively.

Automated Scaling

Configure Application Auto Scaling policies to add or remove task managers dynamically based on real-time load patterns.

Monitoring Essentials: Metrics to Track

Application-Level Metrics

CPUUtilization, MemoryUtilization, and Downtime monitoring (zero downtime means application failure).

Task-Level Metrics

RecordsReceived, RecordsSent, and RecordsLagMax to indicate processing lag and throughput performance.

Checkpoint Metrics

lastCheckpointSize, lastCheckpointDuration, and numberOfFailedCheckpoints for state management monitoring.

Operator-Level Metrics

backPressuredTimeMsPerSecond and idleTimeMsPerSecond to detect processing bottlenecks and inefficiencies.

Focus 2:

Running Flink on Kubernetes (Amazon EKS).

Why Run Flink on Kubernetes & EKS?

Native Orchestration

Kubernetes provides built-in orchestration, automatic scaling, and robust resilience for complex Flink workloads across distributed environments.

Managed Control Plane

Amazon EKS delivers a fully managed, highly secure Kubernetes control plane with seamless AWS service integration and enterprise support.

Cost-Optimised Pipelines

The combination enables cost-efficient, highly available streaming data pipelines with elastic scaling and spot instance optimisation.

Flink Cluster Architecture on Kubernetes

JobManager

Orchestrates job execution and coordinates the entire cluster. Deployed on stable On-Demand instances to ensure consistent control plane availability and decision-making.

TaskManagers

Execute the actual data processing tasks in parallel. Typically run on cost-effective Spot Instances to maximise computational power whilst minimising operational costs.

Operator Control Loop

Continuously monitors FlinkDeployment Custom Resources to maintain desired cluster state, handling failures and scaling events automatically.

Setting Up Flink on EKS: Key Steps

EKS Cluster Creation

Provision EKS cluster with multiple node groups combining On-Demand and Spot instances for optimal cost-performance balance.

Container Image Pipeline

Build optimised Flink Docker images with your applications and push to Amazon ECR for secure, scalable container distribution.

IAM Configuration

Configure fine-grained IAM roles and policies enabling secure access to S3 for checkpoints and Kinesis for data ingestion.

Operator Deployment

Deploy Flink Kubernetes Operator and FlinkDeployment Custom Resources using Helm charts or kubectl for declarative management.


THANK YOU