

Service Name

Kiro AI IDE

Service Overview

Kiro is AWS's native AI-powered development environment and integrated development environment (IDE) that enables developers to build applications using natural language specifications and AI assistance. It provides a comprehensive development experience with integrated access to AWS services, documentation, and real-time AI assistance. Kiro excels at contextual code analysis, infrastructure code generation, visual project management, and collaborative development workflows. The platform supports both CLI and IDE interfaces, leveraging the Model Context Protocol (MCP) to connect with specialized servers and extend capabilities with domain-specific knowledge and functionality.

Key Use Cases

- **Cloud modernization and application migration:** Automating assessment, planning, and implementation phases for migrating legacy applications to AWS
- **Infrastructure optimization:** Finding cost optimization opportunities and automatically applying recommendations to Infrastructure as Code
- **AI agent development:** Building and deploying production-ready AI agents with integrated Amazon Bedrock AgentCore support
- **Database development:** Assisting SQL Server professionals with query optimization, schema analysis, and performance tuning
- **Architecture diagram generation:** Creating AWS architecture diagrams using natural language prompts and MCP servers
- **Citizen development:** Enabling business users to build applications using natural language without traditional coding skills

Features

- **AI-Powered Development:** Natural language code generation and contextual analysis with real-time AI assistance
- **Model Context Protocol Integration:** Connects with specialized MCP servers for extended domain-specific capabilities
- **Infrastructure Code Generation:** Automated generation of Infrastructure as Code with AWS best practices
- **Visual Project Management:** Comprehensive project visualization and management capabilities
- **Cost Optimization:** Automated identification and application of cost optimization recommendations

- **Collaborative Workflows:** Multi-user collaboration features for development teams
- **Spec-Driven Development:** Requirements gathering, design, implementation planning, and execution workflow

Value Proposition

Kiro transforms traditional development workflows by enabling natural language programming and automated code generation, significantly reducing development time from weeks to hours. The platform integrates seamlessly with AWS services and follows AWS best practices automatically, ensuring secure and optimized deployments:

- Key advantages include citizen development capabilities that allow business users to build applications without coding expertise, automated cloud modernization that reduces assessment time from weeks to days, and comprehensive AI assistance that handles complex infrastructure patterns
- Kiro's spec-driven development methodology provides structured requirements, comprehensive design, systematic implementation, and built-in quality through testing and validation, making it ideal for organizations seeking rapid application development and modernization

Service Integration

Kiro integrates extensively with core AWS services through its MCP architecture and built-in connectors. Primary integrations include Amazon Bedrock AgentCore for AI agent runtime and deployment, AWS cost management tools for optimization recommendations, Amazon CloudWatch for monitoring and observability, Amazon RDS and Aurora for database development assistance, and AWS Lambda for serverless application development:

- The platform also connects with AWS documentation services, AWS Pricing API for cost analysis, Amazon S3 for storage management, and AWS Identity Center for authentication
- Kiro works with Infrastructure as Code tools like AWS CDK and CloudFormation, and supports integration with AWS Transform for application modernization and migration workflows

Onboarding and Offboarding

Customers can get started with Kiro by downloading the Kiro CLI or accessing the Kiro IDE through AWS. Initial setup requires configuring AWS credentials and setting up MCP servers for extended functionality:

- The platform offers multiple entry points including direct download, integration with existing IDEs, and web-based access

- Getting started involves installing required dependencies (Python 3.10, specific tools based on use case), configuring MCP servers in the settings file, and connecting to AWS services
- Kiro provides guided tutorials and documentation for common workflows including cloud modernization, cost optimization, and AI agent development
- The initial setup typically takes just a few hours, and users can begin with pilot projects to familiarize themselves with the agentic development approach

Getting Started Guide: Based on available documentation, specific getting started guide URLs are not explicitly provided in the search results, but users are directed to AWS documentation and blog posts for setup instructions

Data Removal

Offboarding information for Kiro is not explicitly detailed in the available documentation. As an AI development environment, data removal would likely involve clearing project workspaces, disconnecting MCP server integrations, removing AWS credential configurations, and deleting any locally stored project artifacts. Users should follow standard practices for removing development environment data and configurations.

Backup/Restore and Disaster Recovery

Backup and disaster recovery capabilities for Kiro are not explicitly documented in the available information. As a development environment, backup considerations would primarily focus on project workspaces, configuration files, and integration settings. Users would rely on standard development practices for code versioning and project backup.

Backup Capabilities

Specific backup capabilities for Kiro are not detailed in the available documentation. The platform likely relies on standard development environment backup practices and integration with version control systems for project preservation.

Disaster Recovery

Disaster recovery support for Kiro is not explicitly documented. As a development environment, disaster recovery would involve standard practices for development tool restoration and reconfiguration of MCP integrations and AWS connections.

Recovery Objectives

Specific RPO and RTO objectives for Kiro are not documented in the available information. Recovery objectives would depend on the underlying infrastructure hosting the development environment and standard development workflow recovery practices.

Service Constraints

Service Quotas: Service quotas documentation specific to Kiro is not available in the search results

FAQ: FAQ documentation specific to Kiro is not available in the search results

Technical Requirements

Service Documentation: Based on search results, Kiro documentation is primarily available through AWS blog posts and integration guides rather than centralized service documentation

User Guide: Specific user guide links are not provided in the search results, with information distributed across various AWS blog posts and tutorials

API Reference: API reference documentation specific to Kiro is not available in the search results

Pricing Overview

Please see the AWS UK G Cloud 15 Pricing Document accompanying this service in the Digital Marketplace.

Public Pricing: Specific pricing information for Kiro is not explicitly detailed in the search results, though costs may be associated with AI model usage and MCP server operations

Cost Optimization

Kiro offers unique cost optimization features including automated identification of optimization opportunities through MCP integration with AWS cost management tools, automatic application of recommendations to Infrastructure as Code, natural language creation of cost-optimized Service Control Policies, and embedded cost optimization rules in development workflows. The platform can analyze entire infrastructure files and suggest comprehensive optimizations including storage tier improvements, networking enhancements, compute right-sizing, auto-scaling configurations, and CloudWatch Logs retention policies. Kiro's context system allows organizations to embed their cost optimization best practices directly into the development workflow, ensuring every new resource is optimized from deployment.

Savings Considerations

Main cost savings come from dramatically reduced development time, with projects moving from weeks to hours through AI-assisted development and automated code generation. Kiro eliminates manual assessment and planning phases in modernization projects, reducing assessment time from weeks to days:

- The platform's automated optimization recommendations can identify significant cost savings opportunities across EC2 instances, auto-scaling groups, storage configurations, and other AWS resources
- Additional savings include reduced need for specialized development resources through citizen development capabilities, decreased maintenance overhead through automated best practices implementation, and prevention of cost overruns through embedded optimization rules and governance frameworks that are built into the development process from the start