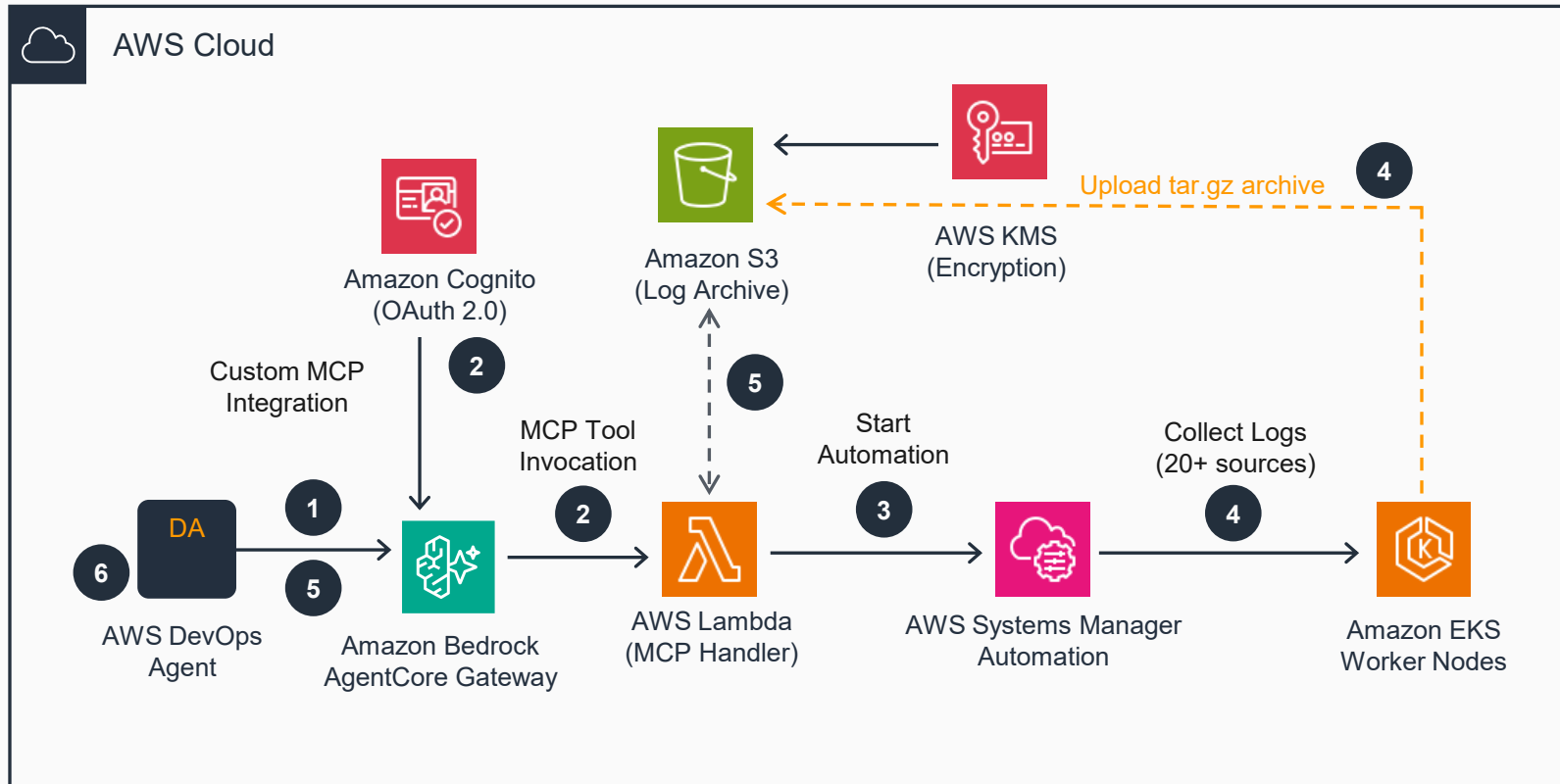


Guidance for EKS Node Diagnostics MCP with DevOps Integration on AWS

This architecture diagram illustrates how to effectively support EKS Node Diagnostics MCP with DevOps Integration on AWS. It shows the key components and their interactions, providing an overview of the architecture's structure and functionality.



- 1 AWS DevOps Agent calls a collect tool with an instance ID. The agent connects to the MCP server through **Amazon Bedrock AgentCore Gateway** over MCP/HTTPS.
- 2 **AgentCore Gateway** authenticates the request using an OAuth 2.0 token from **Amazon Cognito** (client credentials flow) and routes the tool invocation to the registered **AWS Lambda** function.
- 3 The Lambda function (MCP Handler) dispatches an **AWS Systems Manager Automation** execution to the target node, running the AWSSupport-CollectEKSInstanceLogs runbook.
- 4 The SSM Automation runbook collects 20+ log sources on the EKS worker node — kubelet, containerd, iptables, CNI config, route tables, dmesg, sysctl, ENI metadata, IPAMD logs, and more — packages them into an archive and uploads it to **Amazon S3** with **AWS KMS** encryption.
- 5 A processing pipeline extracts the archive, pre-indexes errors with severity classification and stable finding IDs, and provides the structured results back through additional MCP tools.
- 6 AWS DevOps Agent chains tools together (collect → status → errors → search → correlate → summarize) to autonomously investigate and identify root causes without human intervention.