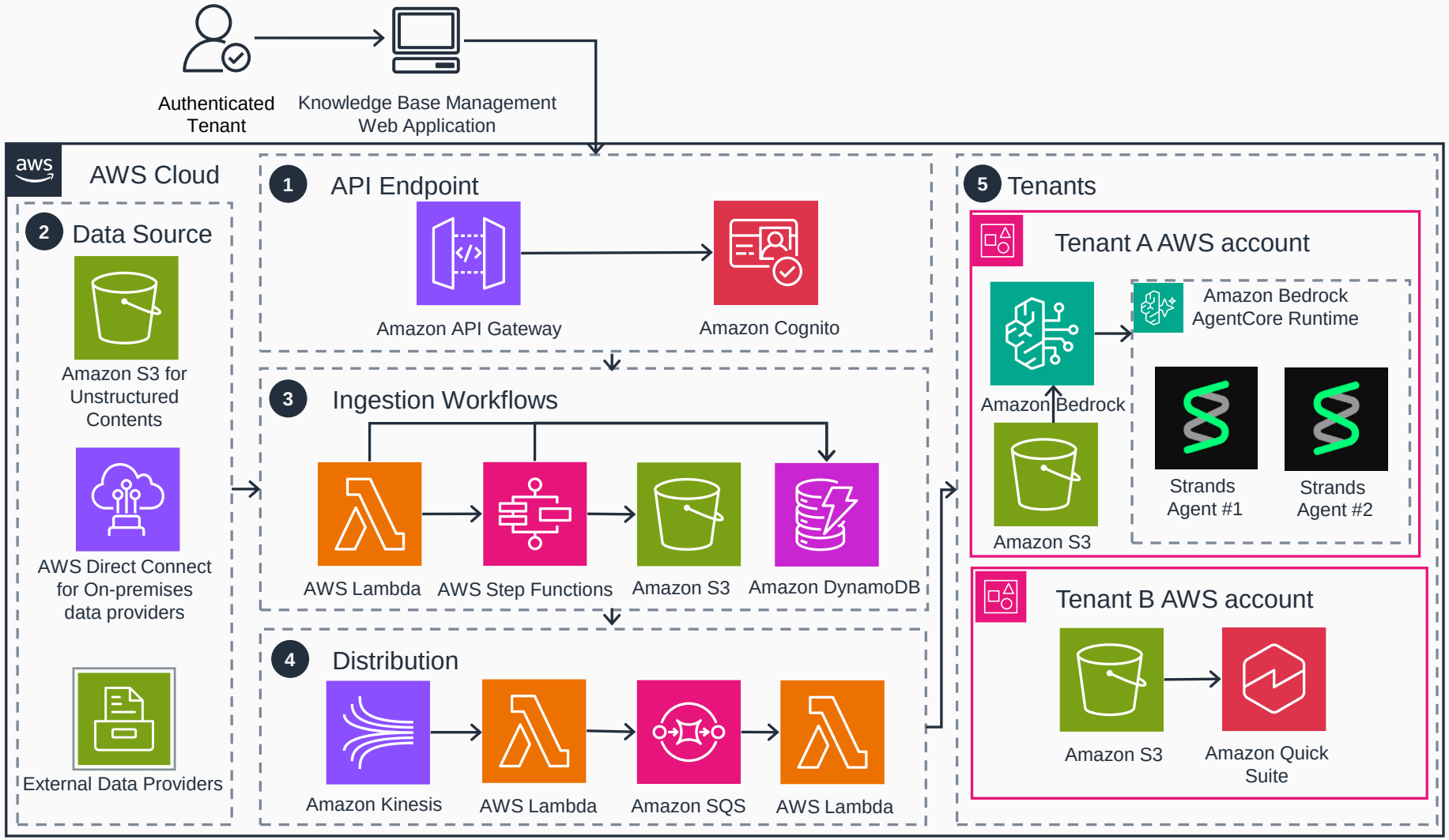
Guidance for Multi-Tenant Knowledge Base Management for scalable RAG Applications on AWS



1. Authenticated tenants access the solution through the **Knowledge Base Management Web Application**, which serves as the primary interface for both submitting content and querying knowledge base data. All requests from the web application are routed through **Amazon API Gateway**, which acts as the secure entry point for all API interactions. **Amazon Cognito** handles identity management, authenticating and authorizing users before any request is processed. Together, API Gateway and Cognito ensure that only verified tenants can perform read and write operations — such as configuring data sources and triggering ingestion workflows through a consistent and secure API layer.

2. **Amazon S3** is the primary content ingestion provider for unstructured contents such as html, pdf, images, etc. AWS Direct Connect enables secure connectivity to on-premises data providers.

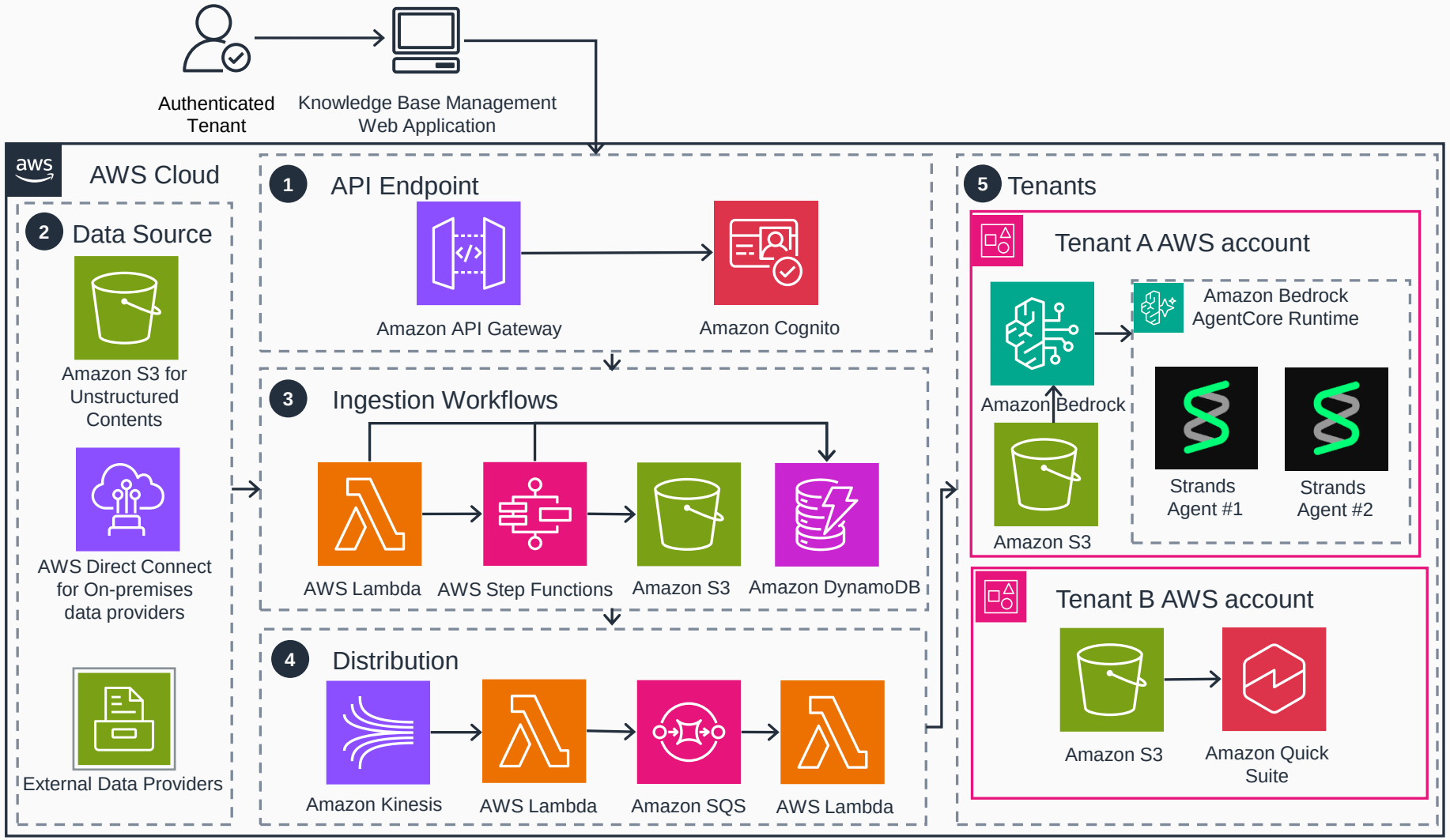
3. **AWS Lambda** serves as the serverless computing service for **AWS Step Functions** tasks. **AWS Step Functions** orchestrates the end-to-end data pipeline, coordinating extract, transform, and distribution steps.

   Amazon S3 stores snapshots of data between intermediate processing steps. This caching strategy serves multiple purposes: audit trail maintenance, governance compliance, performance optimization through reduced reprocessing, and enabling rollback capabilities. **Amazon DynamoDB** used to persist workflow state and tenant configuration data.

Reviewed for technical accuracy March 10, 2026

AWS Reference Architecture

Guidance for Multi-Tenant Knowledge Base Management for scalable RAG Applications on AWS



4 Processed content updates from the ingestion workflows are streamed into **Amazon Kinesis**, which serves as the entry point for the distribution layer. A dedicated **AWS Lambda function** acts as a Kinesis consumer, continuously reading records from the stream. For each content update, this Lambda function looks up the tenant configuration stored in **Amazon DynamoDB** to determine which tenants have subscribed to the content and identifies the corresponding tenant-specific **Amazon SQS queue**. The Lambda then publishes the update to the SQS queue.

Each tenant has its own isolated SQS queue, which decouples the publishing process across tenants. This isolation ensures that a failure in delivering content to Tenant A's account does not affect Tenant B providing fault tolerance and independent retry behavior per tenant. Downstream **AWS Lambda functions** subscribed to each tenant's SQS queue then consume the messages and route the content to the appropriate tenant AWS account for final delivery.

5 **Amazon S3** stores distributed contents as an interim data source for ingestion into target knowledge base systems such as **Amazon Bedrock Knowledge Bases and Amazon Quick Index (part of Amazon Quick Suite)**, an AI-powered workspace that creates a unified knowledge foundation by consolidating documents, files, and application data. **Amazon Bedrock Knowledge Bases** is often used with **Amazon Bedrock AgentCore**, a comprehensive set of enterprise-grade services for deploying and operating AI agents at scale, and can be implemented using frameworks like Strands Agents SDK for agentic use cases.

Reviewed for technical accuracy March 10, 2026

AWS Reference Architecture