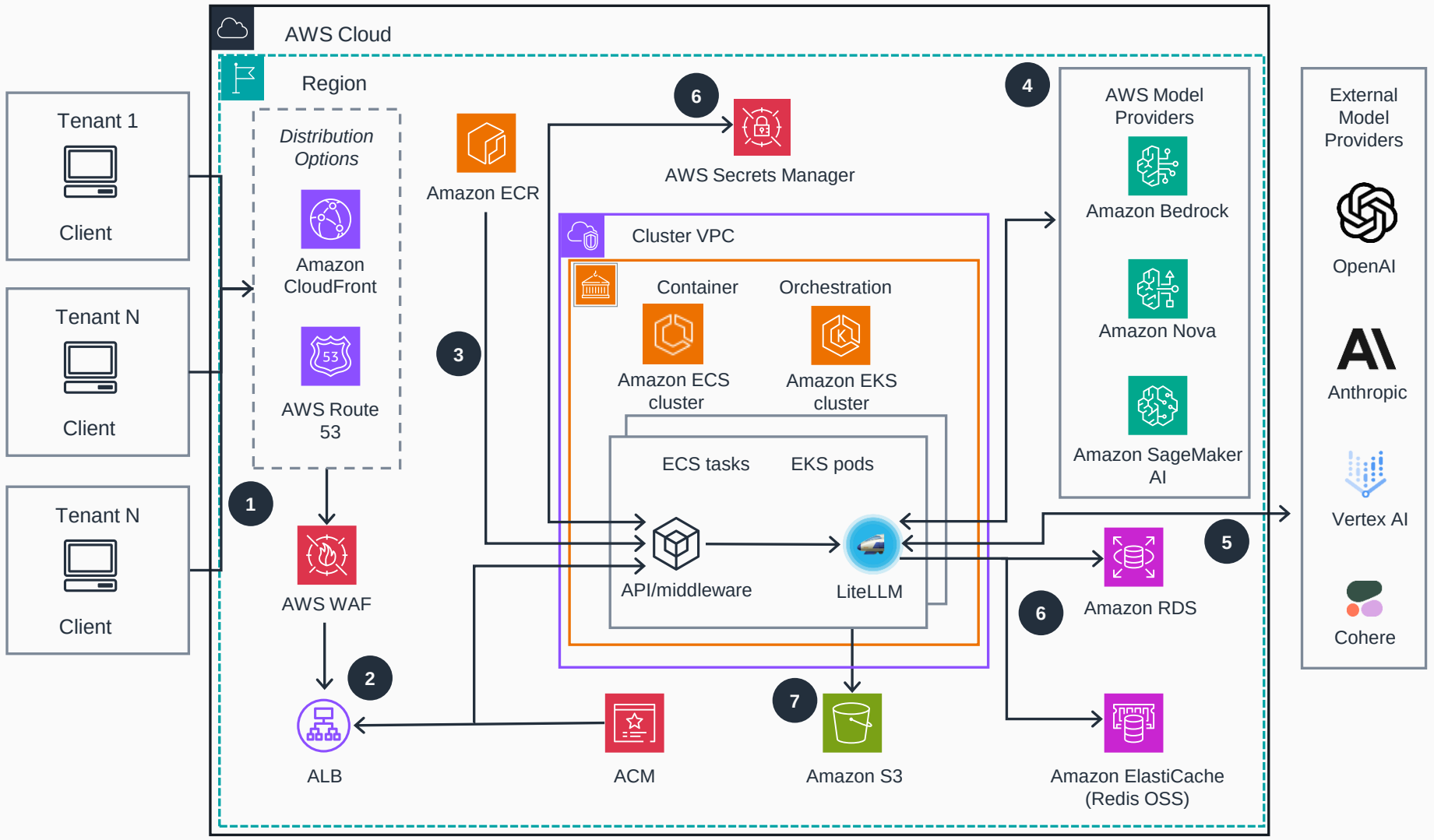
Guidance for Multi-Provider Generative AI Gateway on AWS

This architecture diagram demonstrates how to deploy with Amazon ECS or Amazon EKS container orchestration running on AWS. This slide shows Steps 1-4.



1. Tenants and client applications access the LiteLLM gateway proxy API through the **Amazon Route 53** URL endpoint or **Amazon CloudFront,** which is protected against common web exploits and bots using **AWS WAF**.

2. **AWS WAF** forwards requests to **Application Load Balancer (ALB)** to automatically distribute incoming application traffic to **Amazon Elastic Container Service (Amazon ECS)** tasks or **Amazon Elastic Kubernetes Service (Amazon EKS)** pods running generative AI gateway containers. TLS/SSL encryption secures traffic to the load balancer using a certificate issued by **AWS Certificate Manager (ACM).**

3. Container images for API/middleware and LiteLLM applications are built during guidance deployment and pushed to **Amazon Elastic Container Registry (Amazon ECR)**. They are used for deployment to **Amazon ECS** on **AWS Fargate** or **Amazon EKS** clusters that run these applications as containers in **ECS** tasks or **EKS** pods, respectively. LiteLLM provides a unified application interface for configuration and interacting with LLM providers. The API/middleware integrates natively with **Amazon Bedrock** to enable features not supported by the LiteLLM opensource project.

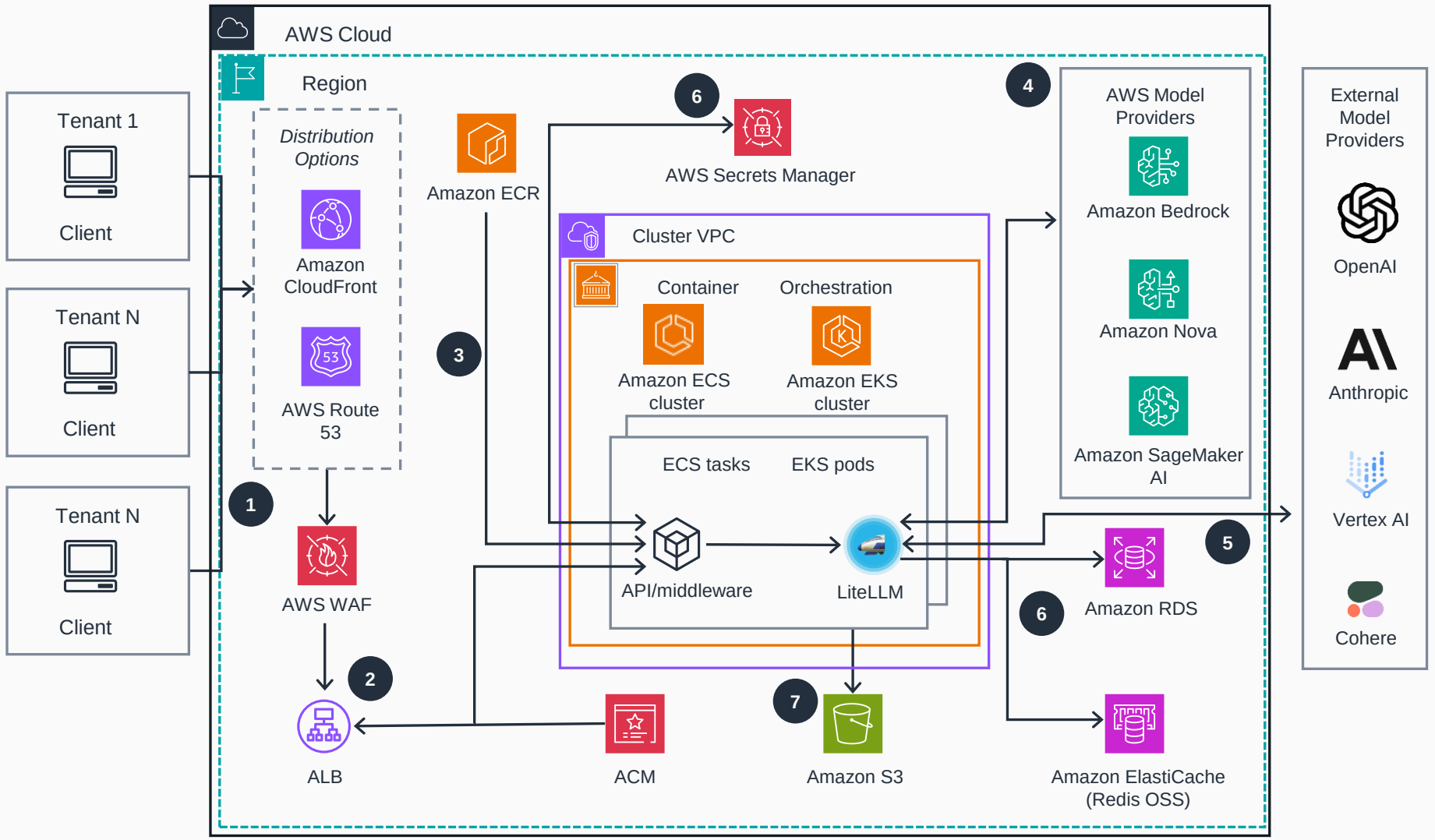
4. Models hosted on **Amazon Bedrock** and **Amazon Nova** provide model access, guardrails, prompt caching, and routing to enhance the AI gateway and additional controls for clients through a unified API. Model access is also available for models deployed on **Amazon SageMaker AI**. Access to required **Amazon Bedrock** models must be properly configured.

Reviewed for technical accuracy July 1, 2025

AWS Reference Architecture

Guidance for Multi-Provider Generative AI Gateway on AWS

This architecture diagram demonstrates how to deploy with Amazon ECS or Amazon EKS container orchestration running on AWS. This slide shows Steps 5-7.



5 External model providers (such as OpenAI, Anthropic, or Vertex AI) are configured using the LiteLLM Admin UI to enable additional model access through LiteLLM's unified application interface. Integrate pre-existing configurations of third-party providers into the gateway using LiteLLM APIs.

6 LiteLLM integrates with **Amazon ElastiCache (Redis OSS), Amazon Relational Database Service (Amazon RDS)**, and **AWS Secrets Manager** services. **Amazon ElastiCache** enables multi-tenant distribution of application settings and prompt caching. **Amazon RDS** enables persistence of virtual API keys and other configuration settings provided by LiteLLM. **Secrets Manager** stores external model provider credentials and other sensitive settings securely.

7 LiteLLM and the API/middleware store application sends logs to the dedicated **Amazon Simple Storage Service (Amazon S3)** storage bucket for troubleshooting and access analysis.

Reviewed for technical accuracy July 1, 2025

AWS Reference Architecture