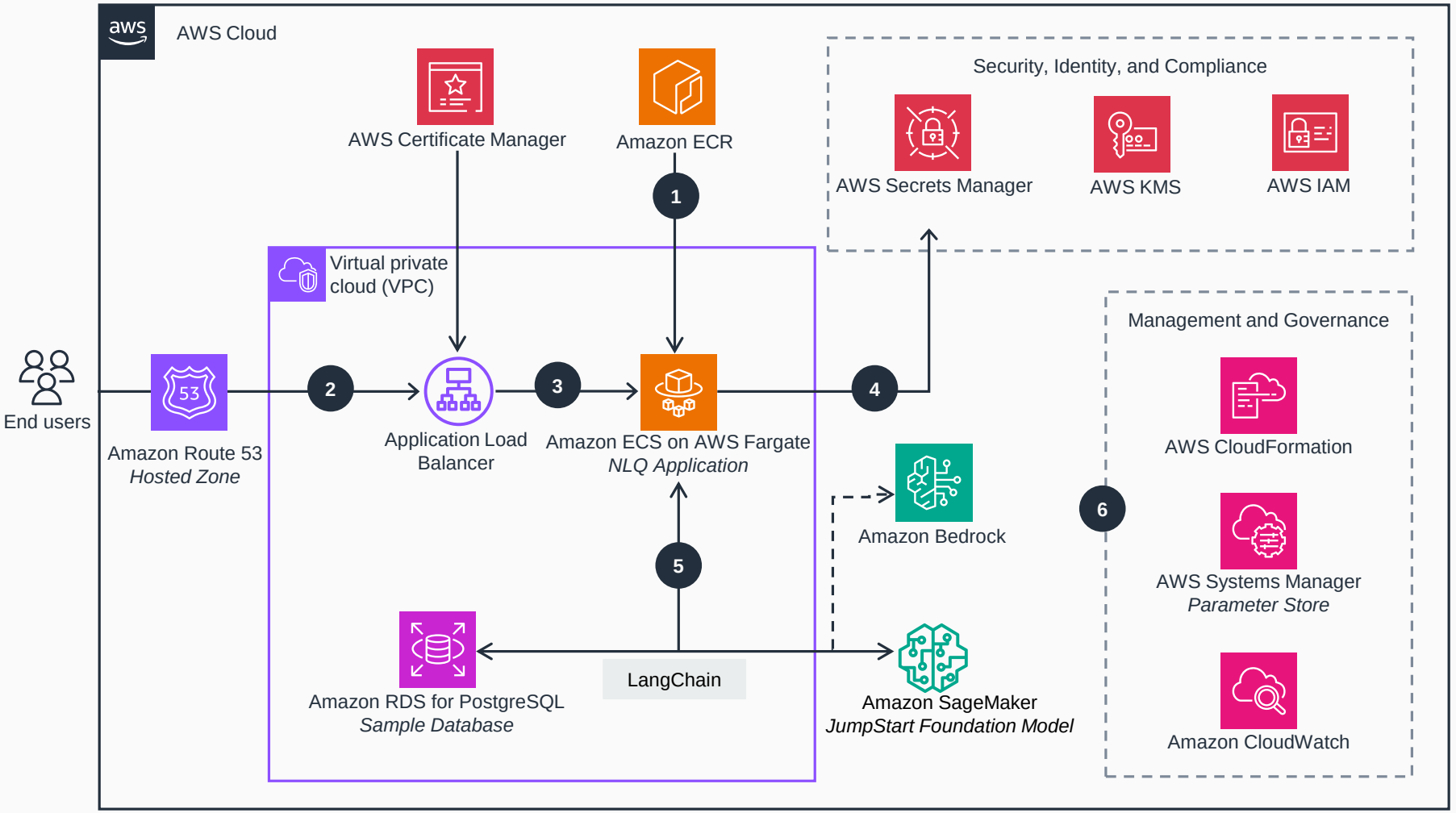
Guidance for Natural Language Queries of Relational Databases on AWS

This architecture diagram demonstrates how to use natural language to query an Amazon RDS for PostgreSQL database. It uses a combination of a generative artificial intelligence (AI) foundation model (FM) along with Streamlit technology integrated with LangChain to quickly build large language models (LLM). It also uses Chroma, an open-source database for embedded vector data. This slide outlines steps 1–4; refer to the next slide for steps 5–6.



1. Create an **Amazon Elastic Container Service (Amazon ECS)** task definition for the natural language query (NLQ) application. Deploy it to **Amazon ECS** on **AWS Fargate**. The Docker image for the task is retrieved from the **Amazon Elastic Container Registry (Amazon ECR)**.

2. End-user requests are received by an Application Load Balancer (ALB) using HTTP. Access is controlled by the IP address using security group inbound (ingress) rules.*

3. End-user requests are forwarded from the ALB to the NLQ application, which runs on **Amazon ECS** using the **Fargate** launch type. The NLQ application's containerized workloads expose their web-based user interface (UI), built with Streamlit, on port 8501.

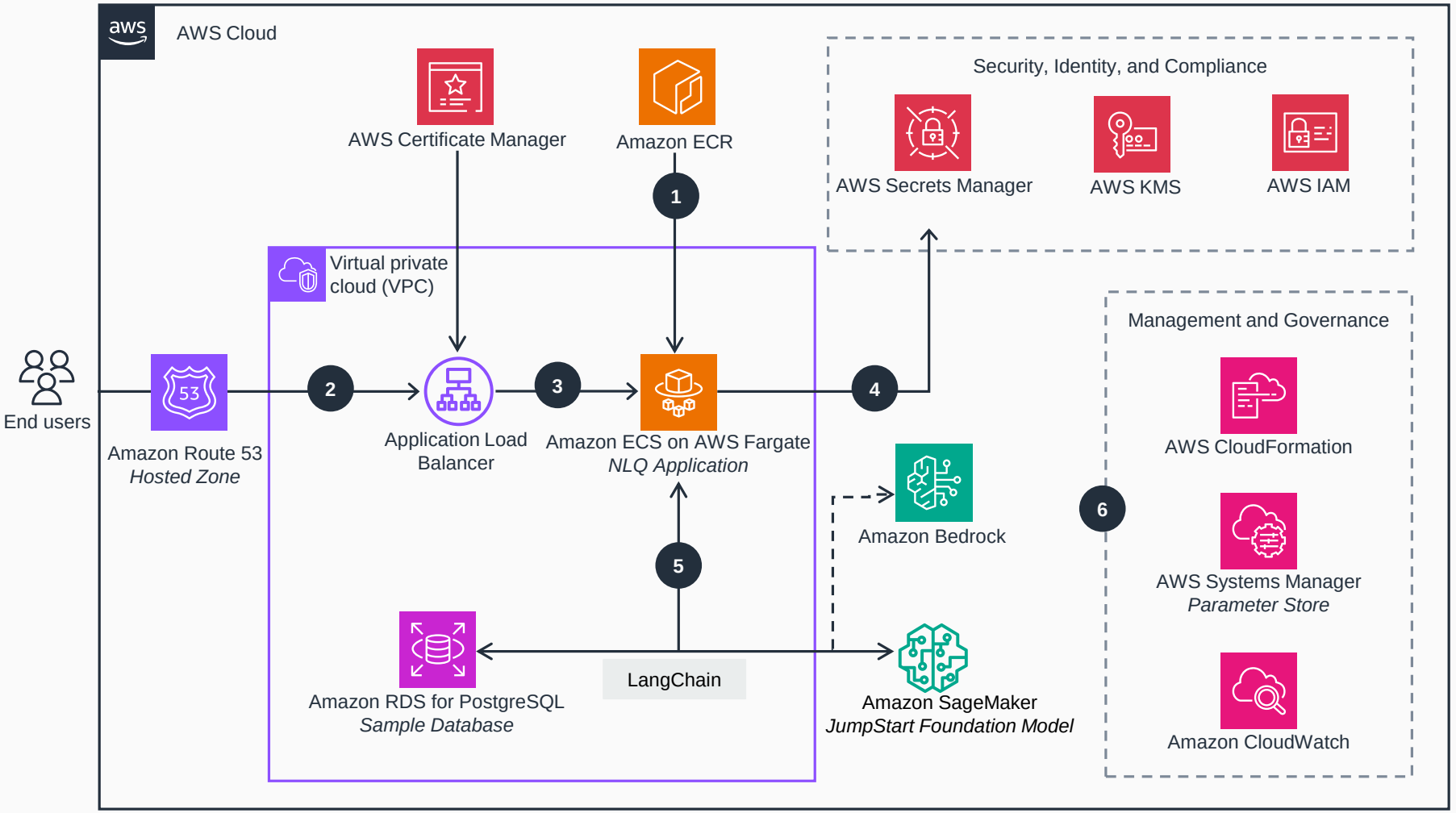
4. The NLQ application retrieves credentials for **Amazon RDS for PostgreSQL** from **AWS Secrets Manager**. **Secrets Manager** uses an **AWS Key Management Service (AWS KMS)** key to decrypt the credentials. Virtual private cloud (VPC) endpoints are used to keep traffic between the VPC and the services on the AWS network.

Reviewed for technical accuracy March 7, 2024

AWS Reference Architecture

Guidance for Natural Language Queries of Relational Databases on AWS

Steps 5-6



5 The user enters a natural language query through the application's web interface built with Streamlit.

The application, using LangChain's SQLDatabaseChain API, interacts with a large language model (LLM) like **Amazon SageMaker JumpStart** or **Amazon Bedrock**. It also interacts with the PostgreSQL database, hosted on **Amazon RDS**, to process the user's natural language query.

To improve accuracy, LangChain uses in-context learning (also referred to as “few-shot prompting”), passing semantically similar SQL query examples to the FM. LangChain uses Chroma to perform a similarity search of relevant samples based on the end-user’s question. LangChain is able to take the end-user’s natural language questions, formulate a SQL query, and query the **Amazon RDS** database. It returns and translates the results into a natural language answer, which is returned to the end-user.

6 This Guidance uses **AWS CloudFormation** for deploying resources. All parameters are stored either in Parameter Store, a capability of **AWS Systems Manager**, or **Secrets Manager**. The NLQ application centralizes all logs, metrics, and information from other AWS services into **Amazon CloudWatch**.

* AWS security best practices dictate that HTTPS should be used, as opposed to HTTP, to secure data in transit. HTTPS requires an SSL/TLS server certificate. Use **AWS Certificate Manager (ACM)** to manage the certificate, which is loaded onto the ALB. The ALB uses the certificate to terminate the front-end HTTPS connection and then decrypt requests from the end-user. The certificate requires a registered DNS hostname, which can be managed using **Amazon Route 53**.

Reviewed for technical accuracy March 7, 2024

AWS Reference Architecture