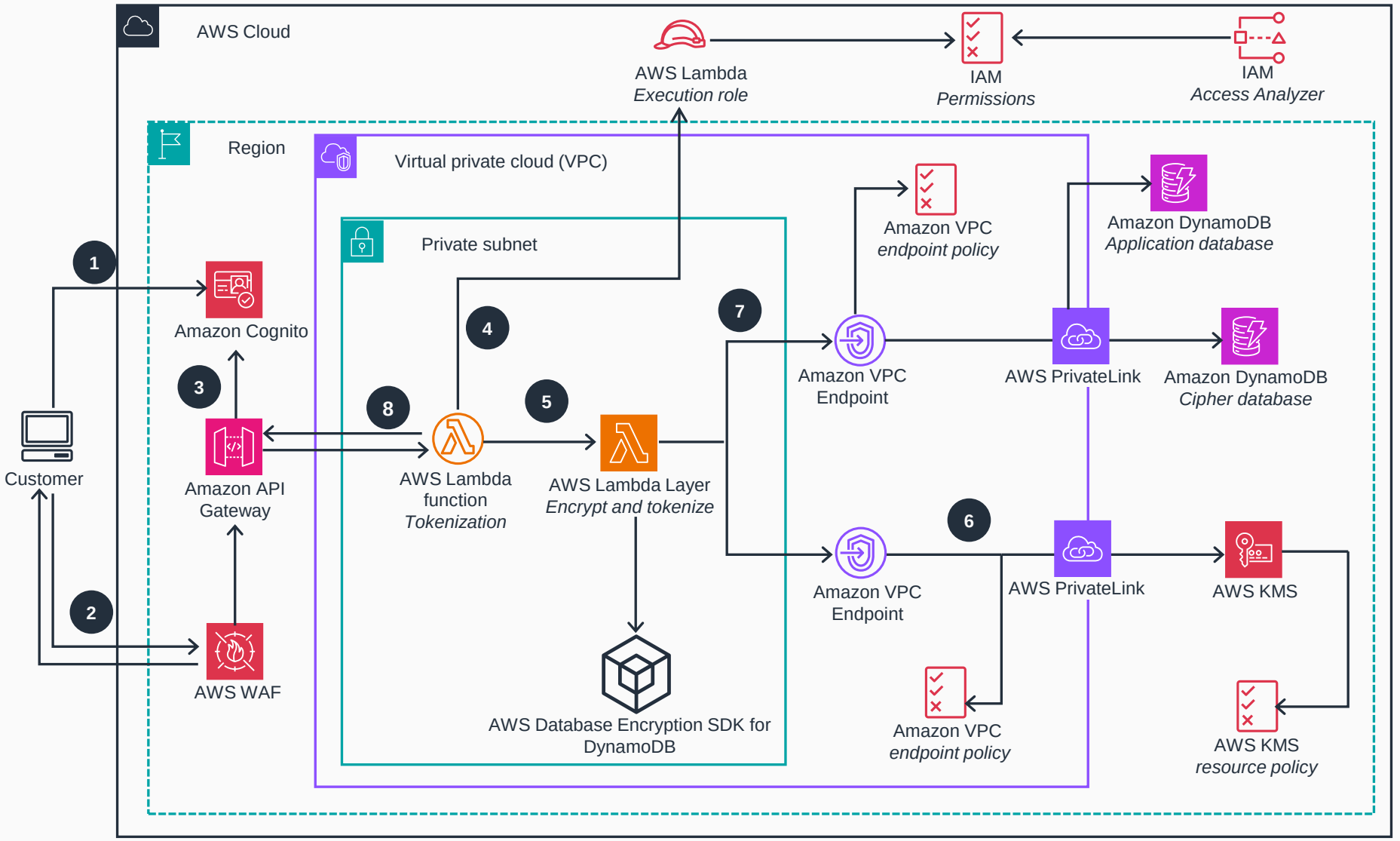
Guidance for Tokenization to Improve Data Security and Reduce Audit Scope on AWS

This architecture diagram shows how you can build a serverless solution to tokenize sensitive data on AWS. With a layered defense design, you can protect the tokenization APIs through the use of an IAM policy, a VPC endpoint policy, and a resource policy.



1. The customer-facing application authenticates with **Amazon Cognito** and obtains an authorization token to access the tokenization APIs.
2. The customer-facing application invokes tokenization APIs using **Amazon API Gateway** with mutual TLS and API keys. The APIs are routed through **AWS WAF** to enforce the intended access.
3. **API Gateway** validates the authorization token and then forwards the requests to the tokenization **AWS Lambda** function.
4. The tokenization **Lambda** function assumes an **AWS Identity and Access Management** (IAM) role to access the **Lambda** layer, the **AWS Key Management Service** (AWS KMS) encryption key, and the **Amazon DynamoDB** databases.
5. The tokenization **Lambda** function uses a verified and version-controlled **Lambda** layer to generate unique tokens for sensitive data.
6. The tokenization **Lambda** layer encrypts the sensitive plaintext using the encryption keys from **AWS KMS**. The connection uses an **Amazon Virtual Private Cloud** (Amazon VPC) endpoint with an endpoint policy to provide additional protection. **AWS KMS** uses a resource policy to validate the permissions for accessing the encryption key.
7. The encrypt and tokenize **Lambda** layer sends the tokenized data to the application database and stores the encrypted text in a cipher database for future retrieval. The connection uses an **Amazon VPC** endpoint with an endpoint policy to provide additional protection. The application database and the cipher database reside in different AWS accounts.
8. The tokenization **Lambda** function returns the tokenized data back to the customer-facing application upon request.

Reviewed for technical accuracy July 23, 2024

AWS Reference Architecture