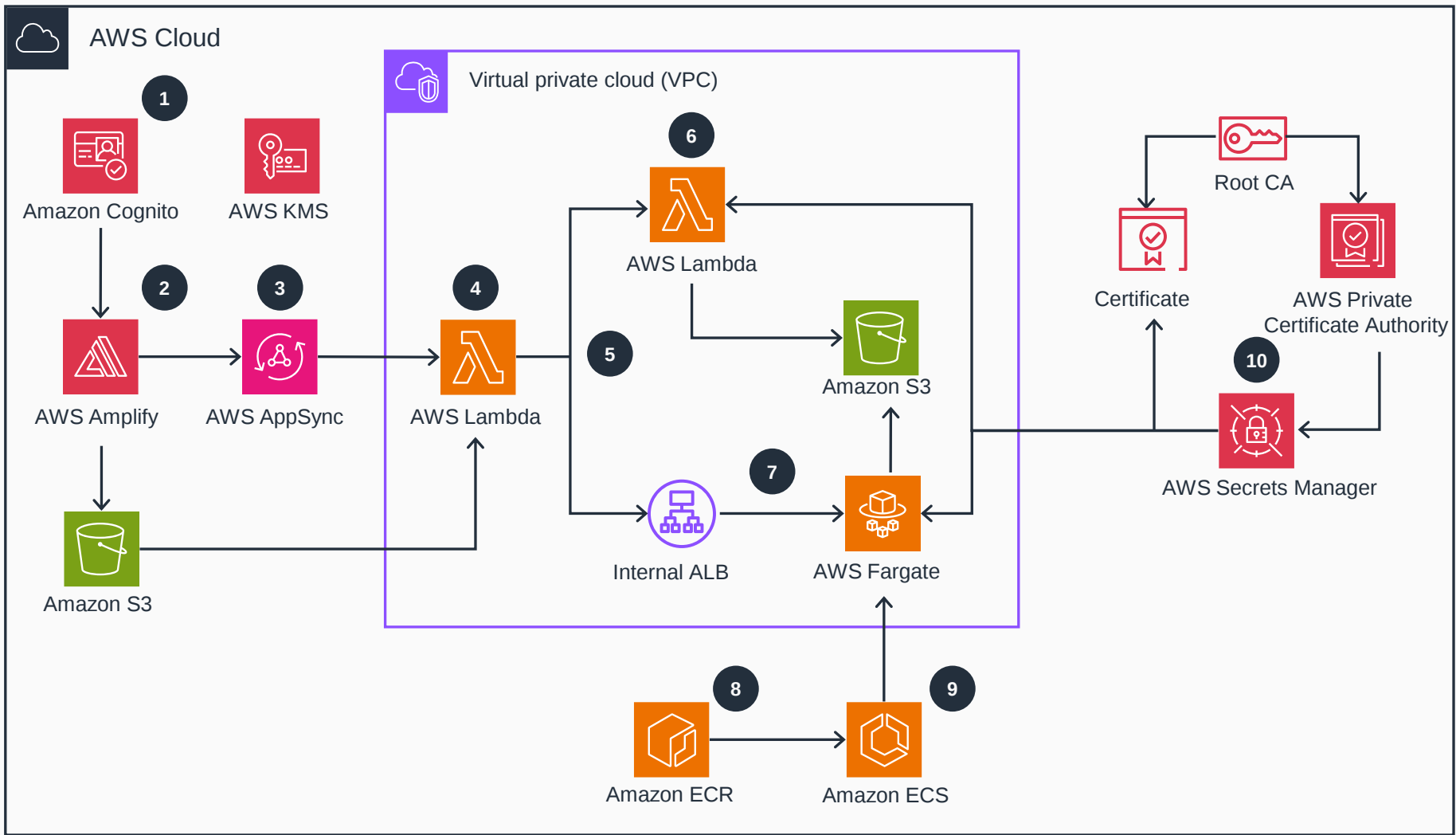


Guidance for Media Provenance with C2PA on AWS

This architecture diagram shows creating, signing, and verifying content authenticity manifests in media workflows on AWS.



- 1 Users access UI with **Amazon Cognito** managed authentication workflow.
- 2 Users upload images, videos, or fragmented MP4s with **AWS Amplify** to an **Amazon Simple Storage Service (Amazon S3)** using **AWS Key Management Service (AWS KMS)** to encrypt and decrypt stored objects.
- 3 **AWS AppSync** provides the API to **Amplify** for the UI.
- 4 **AWS Lambda** routes requests and responses between the frontend and the compute.
- 5 Users select **Lambda** or **AWS Fargate** options to create or extend the C2PA manifests.
- 6 **Lambda** can be used for short tasks such as signing of images.
- 7 **Fargate** can be used for longer tasks such as signing of videos. An **Application Load Balancer (ALB)** exposes the REST API on **Fargate**.
- 8 The container image on **Amazon Elastic Container Registry (Amazon ECR)** contains the signing tool.
- 9 **Amazon Elastic Container Service (Amazon ECS)** pulls the container image from **Amazon ECR**, then **Fargate** runs the containers in a serverless environment.
- 10 **AWS Secrets Manager** securely stores root CA certificates and private keys which are used to sign a claim in a C2PA manifest.

