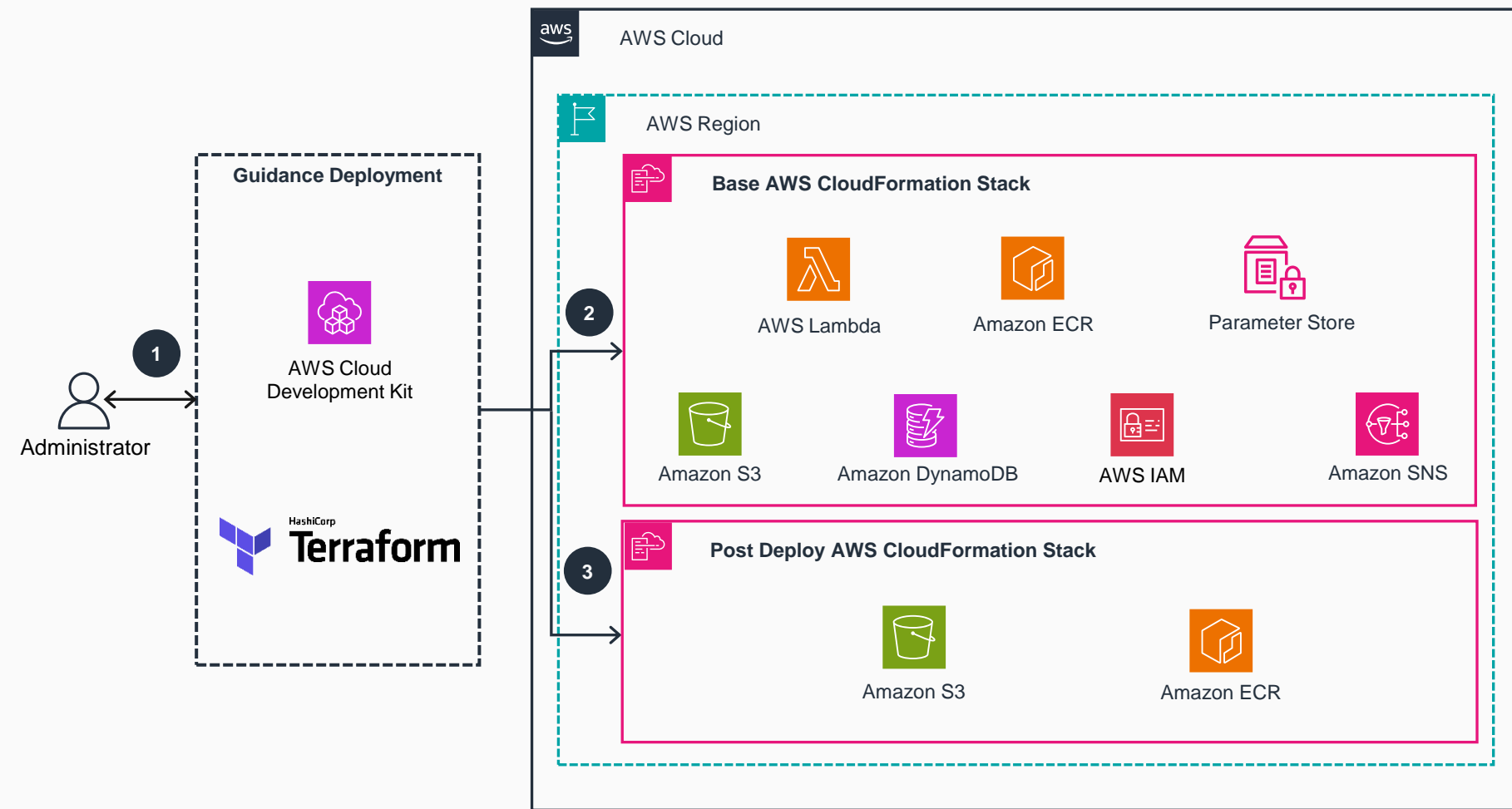# Guidance for Open Source 3D Reconstruction Toolbox for Gaussian Splats on AWS

Event-driven serverless architecture
This architecture diagram shows an automated deployment of event-driven serverless architecture for user accounts.
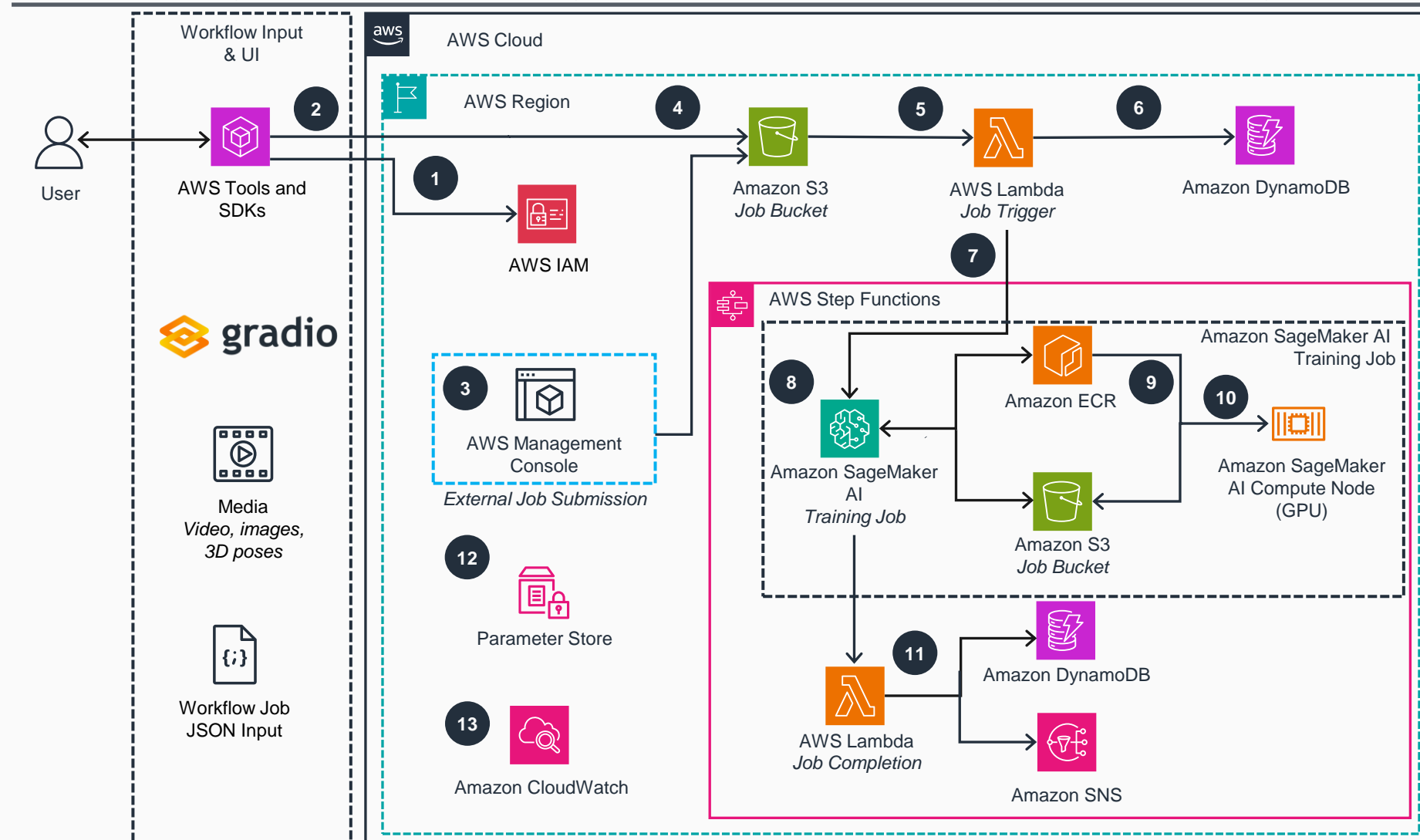

AWS Cloud

AWS Region

Base AWS CloudFormation Stack

AWS Lambda   Amazon ECR   Parameter Store

Amazon S3   Amazon DynamoDB   AWS IAM   Amazon SNS

Post Deploy AWS CloudFormation Stack

Amazon S3   Amazon ECR

Guidance Deployment

AWS Cloud Development Kit

HashiCorp Terraform

Administrator

1. An administrator deploys the Guidance to an AWS account and Region using **AWS Cloud Development Kit (AWS CDK)** or **Terraform**.

2. The Base **AWS CloudFormation** stack to deploy will create all the AWS resources needed to host the Guidance. This includes: an **Amazon Simple Storage Service (Amazon S3)** bucket, **AWS Lambda** functions, an **Amazon DynamoDB** table, necessary **AWS Identity and Access Management (IAM)** permissions, and an **Amazon Elastic Container Registry (Amazon ECR)** image registry. Additionally, it includes an **AWS Step Functions** state machine resource ID in **Parameter Store**, a capability of **AWS Systems Manager**, and it creates an **Amazon Simple Notification Service (SNS)** topic.

3. Once the Base **CloudFormation** stack has been deployed, deploy the Post Deploy **CloudFormation** stack. That stack will build a Docker container and push it to the **Amazon ECR** registry. It will also build and push the pre-processing models used during training, such as for background removal, into the **S3** bucket.

AWS Reference Architecture

Guidance for Open Source 3D Reconstruction Toolbox for Gaussian Splats on AWS

AI-generated open source 3D content

This architecture diagram shows how to enable the generation of realistic 3D content through open source rendering techniques.

Workflow Input & UI

User

AWS Tools and SDKs

gradio

Media
Video, images, 3D poses

Workflow Job JSON Input

AWS Cloud

AWS Region

AWS IAM

AWS Management Console
External Job Submission

Parameter Store

Amazon CloudWatch

Amazon S3
Job Bucket

AWS Lambda
Job Trigger

Amazon DynamoDB

AWS Step Functions

Amazon SageMaker AI
Training Job

Amazon ECR

Amazon SageMaker AI Training Job

Amazon SageMaker AI Compute Node (GPU)

Amazon S3
Job Bucket

AWS Lambda
Job Completion

Amazon DynamoDB

Amazon SNS

1. The user authenticates to **IAM** using **AWS Tools and SDKs**.

2. The input is uploaded to a dedicated **S3** job bucket location. This can be done using a Gradio interface and **AWS Software Development Kit (AWS SDK)**.

3. Optionally, the Guidance supports external job submission by uploading a '.JSON' job configuration file and media files into a designated **S3** job bucket location.

4. The job JSON file uploaded to the **S3** job bucket will trigger an **Amazon SNS** message that will invoke the initialization *Job Trigger* **Lambda** function.

5. The *Job Trigger* **Lambda** function will perform input validation and set appropriate variables for the **Step Functions State Machine**.

6. The workflow job record will be created in the **DynamoDB** job table.

7. The *Job Trigger* **Lambda** function will invoke **Step Functions State Machine** to handle the entire workflow job.

8. An **Amazon SageMaker AI** Training Job will be submitted synchronously using the state machine built-in wait until completion mechanism.

9. The **Amazon ECR** container image and **S3** job bucket model artifacts will be used to deploy a new container on a graphics processing unit (GPU) based compute node. The compute node instance type is determined by the job JSON configuration.

10. The container will run the entire pipeline as an **Amazon SageMaker AI** training job on a GPU compute node.

11. The *Job Completion* **Lambda** function will complete the workflow job by updating the job metadata in **DynamoDB** and using **Amazon SNS** to notify the user through email upon completion.

12. The internal workflow parameters are stored in **Parameter Store** during deployment to decouple the *Job Trigger* **Lambda** function and the **Step Function State Machine**.

13. **Amazon CloudWatch** logs and monitors the training jobs, surfacing possible errors to the user.

AWS Reference Architecture