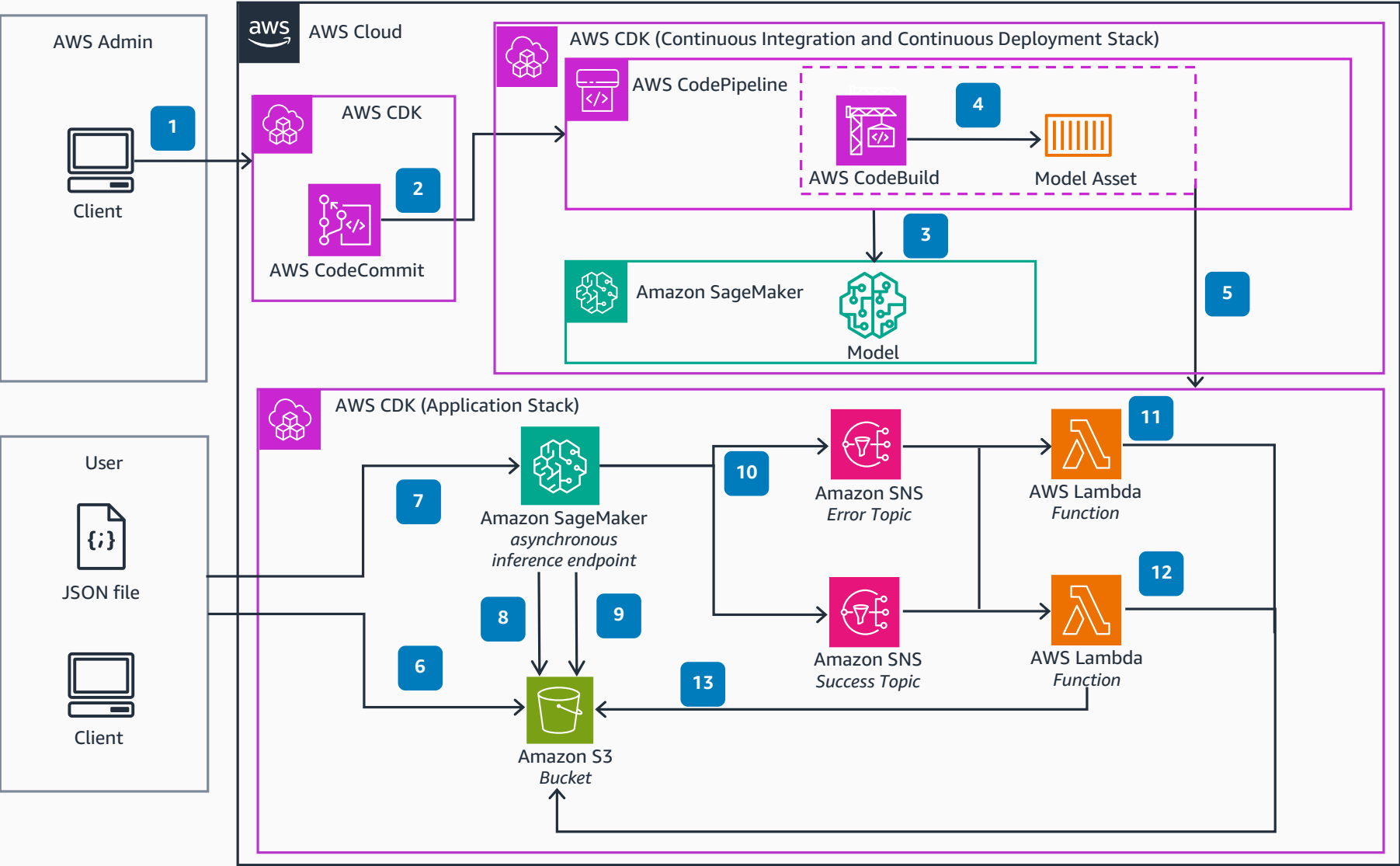


Guidance for Generative AI Deployments using Amazon SageMaker JumpStart

This Guidance demonstrates how to build an asynchronous Amazon SageMaker JumpStart foundation model through an AWS Cloud Development Kit (AWS CDK).



- 1 The AWS Admin deploys the **AWS Cloud Development Kit (AWS CDK)** repository and pipeline stacks, and pushes the project code to **AWS CodeCommit**.
- 2 Once the code is pushed to **CodeCommit**, **AWS CodePipeline** invokes automatically.
- 3 **AWS CodeBuild** downloads the foundational model inference code and the foundational model data from **Amazon SageMaker**.
- 4 **CodeBuild** re-packages the retrieved model inference code and foundational model data into an image generation model to be used with a **SageMaker** endpoint later.
- 5 The **CodePipeline** then deploys the application stack.
- 6 The client or user can upload their model input (example - an image generation prompt) with parameters as a JSON file to an **Amazon Simple Storage Service (Amazon S3)** bucket.
- 7 The user invokes the asynchronous **SageMaker** endpoint.
- 8 The **SageMaker** endpoint scales up inside its Application Auto Scaling group by starting at least one inference compute instance. It reads the input file from the **Amazon S3** bucket.
- 9 The endpoint generates the result according to the input, and stores the raw output in a JSON file in the same **Amazon S3** bucket.
- 10 The **SageMaker** endpoint sends the completed result of the operation to either a Success or Error **Amazon Simple Notification Service (Amazon SNS)** topic.
- 11 The result from either topic is stored in the **Amazon S3** bucket through an **AWS Lambda** function subscribed to the topics for easy state tracking by the user.
- 12 In case the completion was successful, another **Lambda** function subscribed to the Success topic performs post-processing on the result from the **Amazon S3** bucket.
- 13 **Lambda** generates the post-processed result (example - converting JSON RGB values into a PNG image file) and stores it in the **Amazon S3** bucket