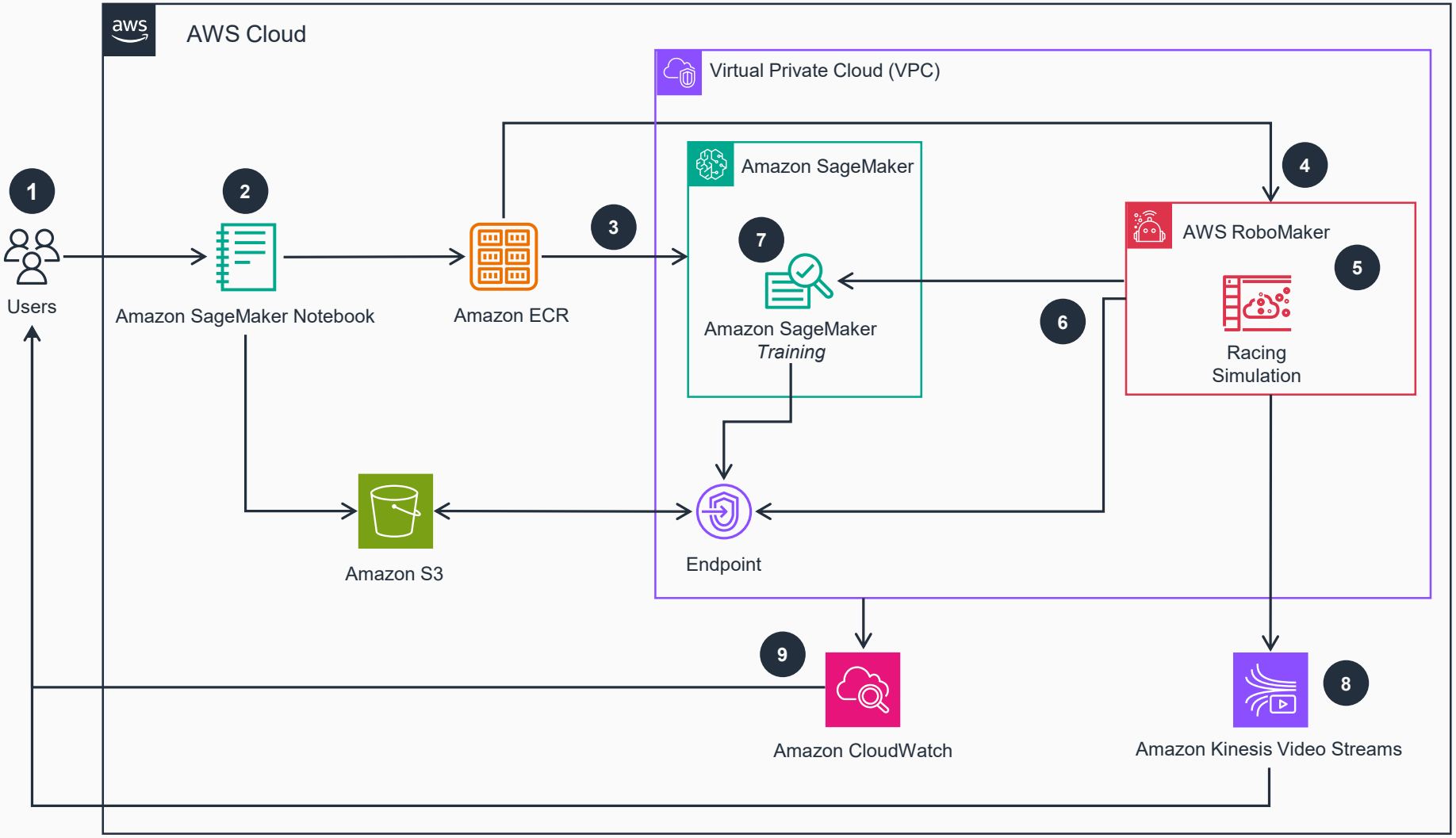
Guidance for Training an AWS DeepRacer Model Using Amazon SageMaker

This architecture diagram is intended for software developers, showing how they can use an Amazon SageMaker Notebook instance to directly train and evaluate AWS DeepRacer models with full control. This includes augmenting the simulation environment, manipulating inputs to the neural network, modifying neural network architecture, running distributed rollouts, and debugging their model.



1. The User logs into their AWS account, creates an **Amazon SageMaker Notebook** to train a Reinforcement Learning (RL) model.
2. **SageMaker Notebook** stores all files required for the training and evaluation jobs in **Amazon Simple Storage Service** (Amazon S3) and as images in **Amazon Elastic Container Registry** (Amazon ECR).
3. **Amazon SageMaker** downloads images from **Amazon ECR** and starts a model training job.
4. **AWS RoboMaker** downloads images from **Amazon ECR** and creates a racing simulation environment for **AWS DeepRacer**.
5. **RoboMaker** starts data generation for model training. Training data generated by **RoboMaker** is a collection of tuples comprising of agent initial state, action, new state, reward.
6. **RoboMaker** sends pre configured batches of these tuples, called iterations, to **Amazon S3** and writes a key to a **SageMaker** container.
7. **SageMaker** training picks up the data file from **Amazon S3** based on the key, and trains the model on this dataset. **SageMaker** uploads the next version of the model to **Amazon S3** that is picked up by **RoboMaker**.
   Steps 5 to 7 are iterative, ending when the training job duration ends. Training job duration can be configured in **SageMaker Notebook** code using the parameter job_duration_in_seconds.
8. **RoboMaker** streams live training and evaluation jobs to **Amazon Kinesis Video Streams**. Users can view model training and evaluation in real time on **Kinesis Video Streams**.
9. All training logs, evaluation logs, service calls, and operational metrics can be viewed in the **Amazon CloudWatch** console.

Reviewed for technical accuracy October 23, 2023

AWS Reference Architecture