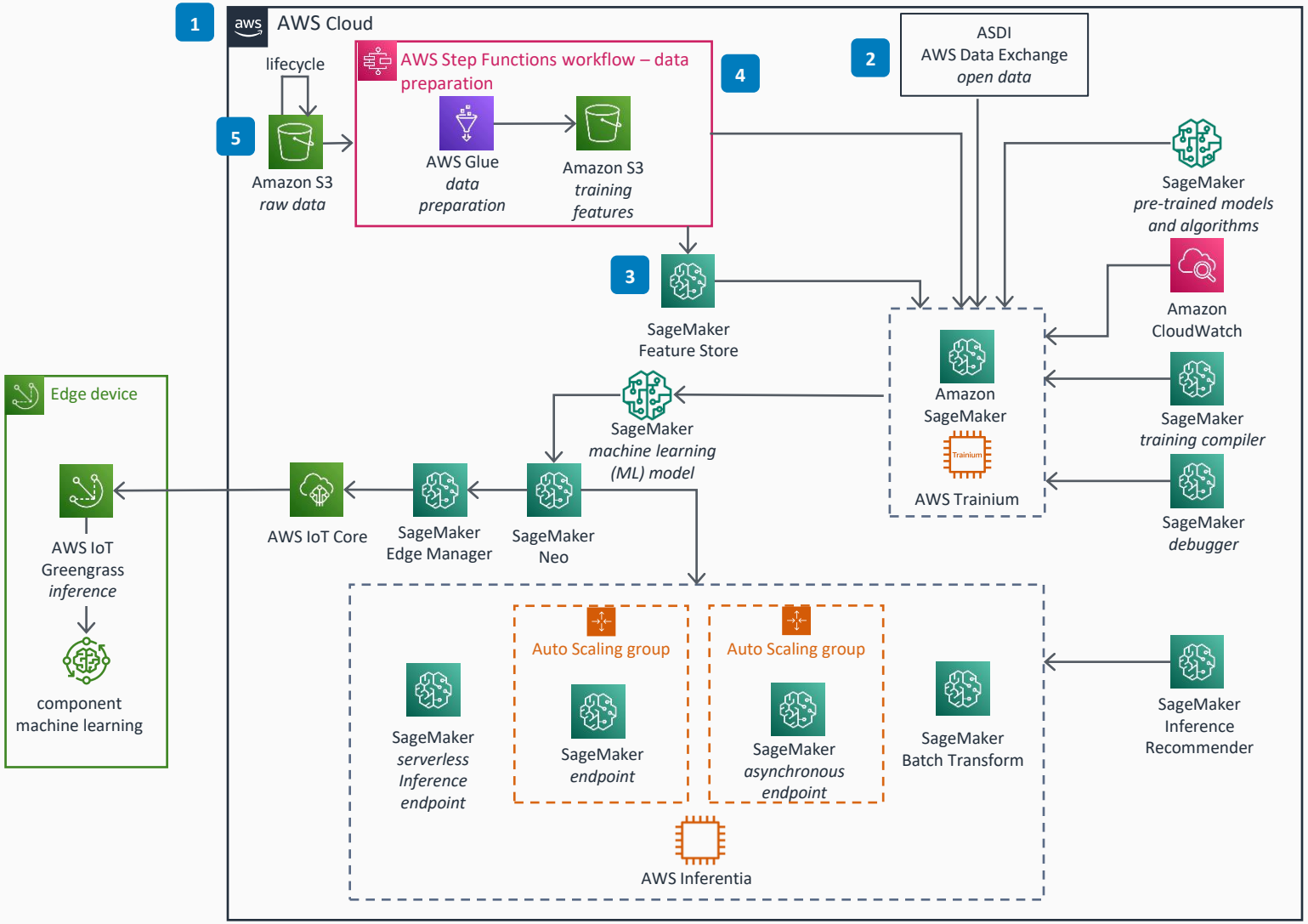
Guidance for Optimizing Deep Learning Workloads for Sustainability on AWS

Data processing

Use this reference architecture to reduce the environmental impact of your deep learning workloads.



1. Select an AWS Region with sustainable energy sources. When regulations and legal aspects allow, choose Regions near Amazon renewable energy projects and Regions where the grid has low published carbon intensity to host your data and workloads. When selecting a Region, try to minimize data movement across networks; store your data close to your producers and train your models close to your data.

2. Evaluate whether you can avoid data processing by using existing publicly available datasets such as **AWS Data Exchange** and **Open Data on AWS**, which includes the Amazon Sustainability Data Initiative (ASDI). They offer weather and climate datasets, satellite imagery, and air quality or energy data, among others. Using these curated datasets avoids duplicating the compute and storage resources needed to download the data from the providers, store it in the cloud, organize, and clean it.

3. For internal data, you can also reduce duplication and rerun of feature engineering code across teams and projects by using **Amazon SageMaker Feature Store**.

4. Adopt a serverless architecture for your data pipeline so it only provisions resources when work needs to be done. Use **AWS Glue** and **AWS Step Functions** for data ingestion and preprocessing, so you are not maintaining compute infrastructure 24/7. **Step Functions** can orchestrate **AWS Glue** jobs to create event-based serverless Extract, Transform, Load/Extract, Load, and Transform (ETL/ELT) pipelines.

5. Use the appropriate **Amazon Simple Storage Service** (Amazon S3) storage tier to reduce the carbon impact of your workload. Use energy-efficient, archival-class storage for infrequently accessed data. If you can easily recreate an infrequently accessed dataset, use the **Amazon S3** One Zone-IA class to minimize the total data stored.

   Manage the lifecycle of all your data and automatically enforce deletion timelines to minimize the total storage requirements of your workload using **Amazon S3** Lifecycle policies. The **S3** Intelligent-Tiering storage class automatically moves your data to the most sustainable access tier when access patterns change. Define data retention periods that support your sustainability goals while meeting your business requirements, not exceeding them.

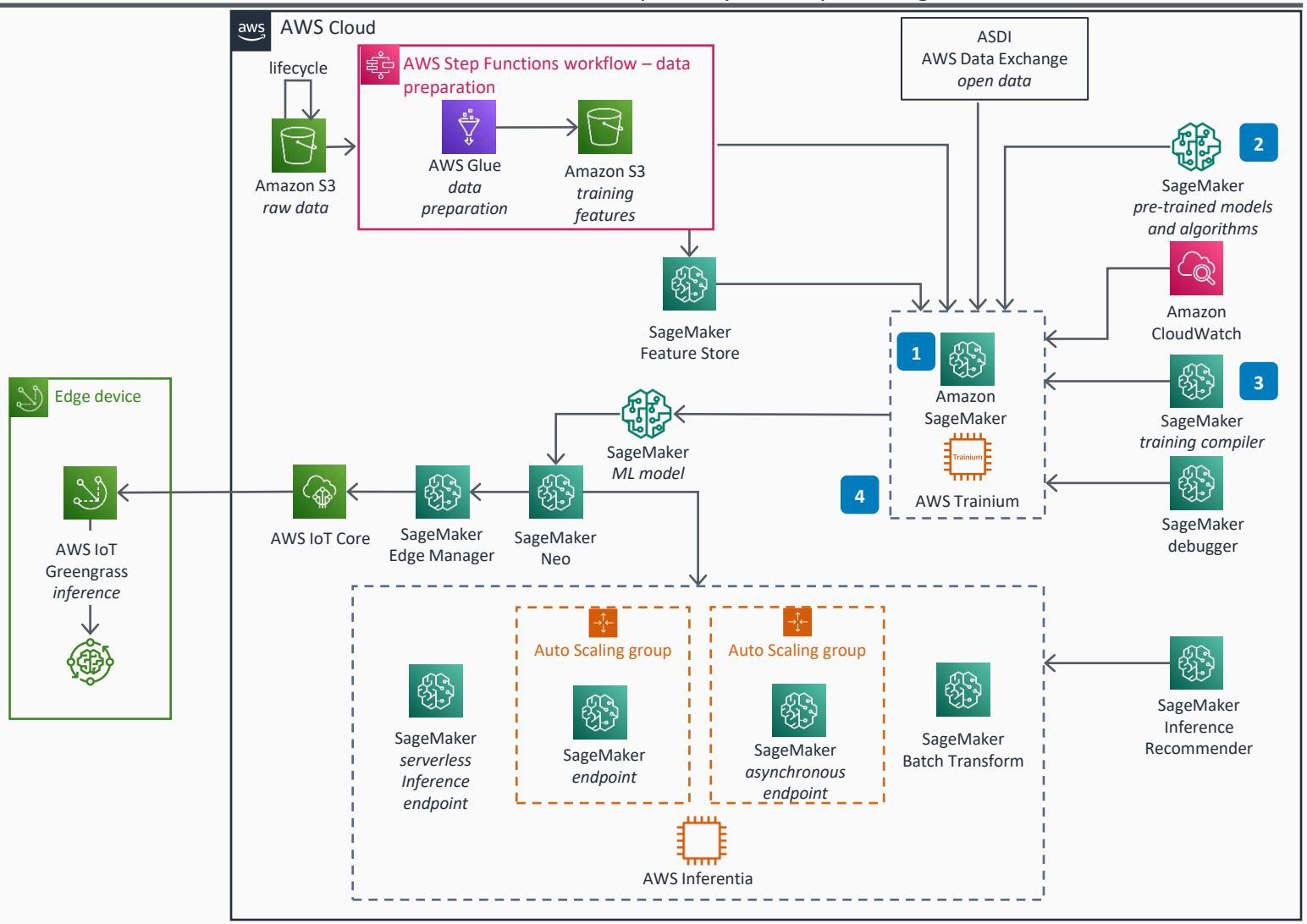
Reviewed for technical accuracy October 4, 2022

AWS Reference Architecture

Guidance for Optimizing Deep Learning Workloads for Sustainability on AWS

Model building

Use this reference architecture to reduce the environmental impact of your deep learning workloads.



1 Define acceptable performance criteria. When you build an ML model, you'll likely need to make trade-offs between your model's accuracy and its carbon footprint. Establish performance criteria that support your sustainability goals while meeting your business requirements, not exceeding them.

2 Evaluate if you can use pre-existing datasets, algorithms, or models. AWS Marketplace offers over 1,400 ML-related assets that customers can subscribe to. You can also fine-tune an existing model such as those available on Hugging Face, or use a pre-trained model from **SageMaker JumpStart**. Using pre-trained models can reduce the resources you need for data preparation and model training.

Try to find simplified versions of algorithms. This will help you use less resources to achieve a similar outcome. For example, DistilBERT, a distilled version of BERT, has 40% fewer parameters, runs 60% faster, and preserves 97% of BERT's performance.

Consider techniques to avoid training a model from scratch. Transfer learning (use a pre-trained source model and reuse it as the starting point for a second task) or incremental training (use artifacts from an existing model on an expanded dataset to train a new model).

3 Use **Amazon SageMaker** Training Compiler to compile your Deep Learning models from their high-level language representation to hardware-optimized instructions to reduce training time. This can speed up training of Deep Learning models by up to 50% by more efficiently using **SageMaker** graphics processing unit (GPU) instances.

4 Automate the ML environment. When building your model, use Lifecycle Configuration Scripts to automatically stop idle **SageMaker Notebook** instances. If you are using **SageMaker Studio**, install the auto-shutdown Jupyter extension to detect and stop idle resources. Use the fully managed training process provided by **SageMaker** to automatically launch training instances and shut them down as soon as the training job is complete. This minimizes idle compute resources, and limits the environmental impact of your training job.

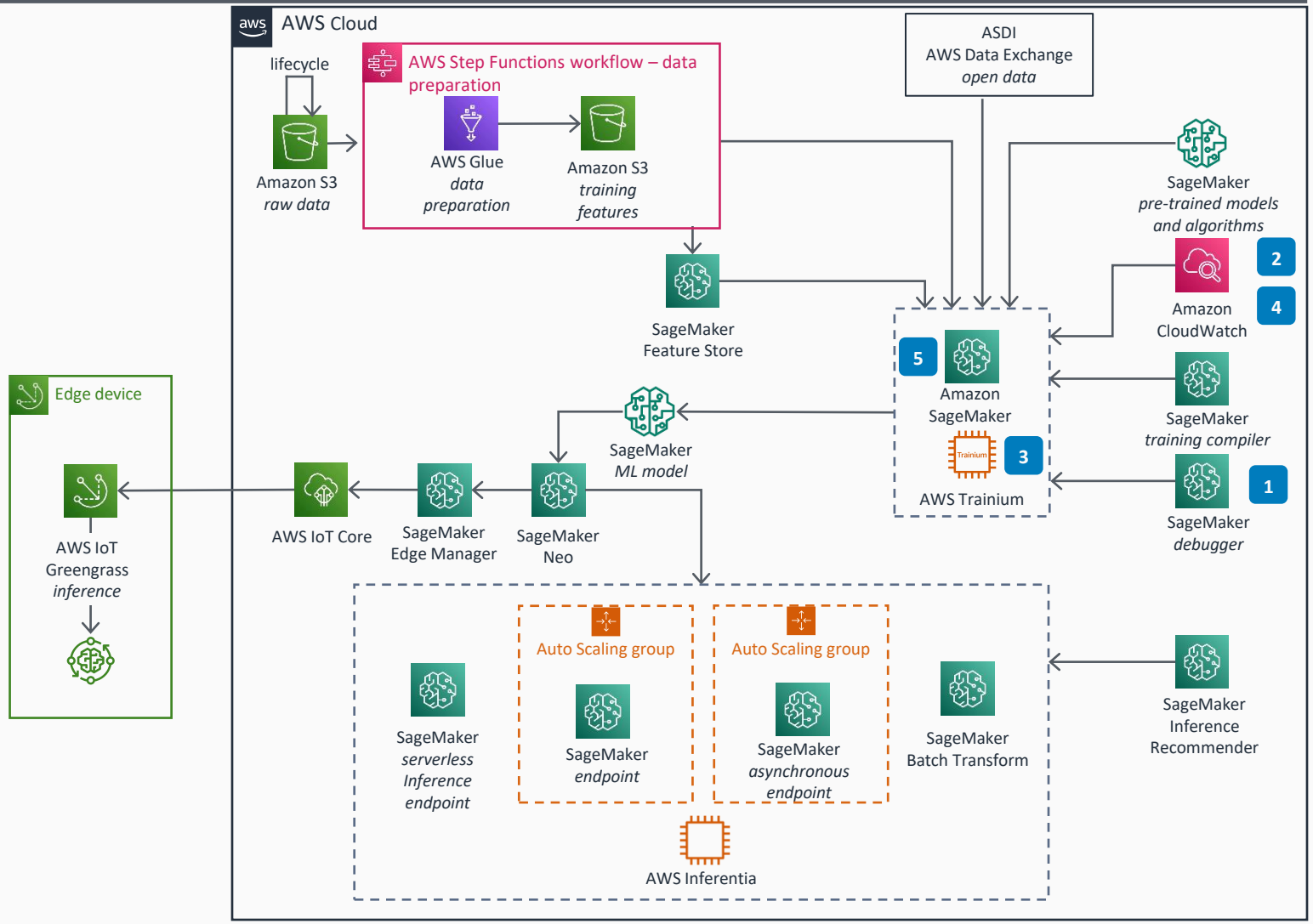
Reviewed for technical accuracy October 4, 2022

AWS Reference Architecture

Guidance for Optimizing Deep Learning Workloads for Sustainability on AWS

Model Training

Use this reference architecture to reduce the environmental impact of your deep learning workloads.



1. Use **SageMaker Debugger** to identify training problems. With built-in rules such as system bottlenecks, overfitting, and saturated activation functions, **SageMaker Debugger** can monitor your training jobs and automatically stop them as soon as it detects a bug, which helps you avoid unnecessary carbon emissions.

2. Right-size your training jobs with **Amazon CloudWatch** metrics that monitor the utilization of resources such as CPU, GPU, memory, and disk utilization. **SageMaker Debugger** also provides profiler capabilities to detect under-utilization of system resources and right-size your training environment. This helps avoid unnecessary carbon emissions.

3. Use **AWS Trainium** to train your deep learning workloads. It is expected to be the most energy efficient processor offered by AWS for this purpose.

4. Consider Managed Spot Training, which takes advantage of unused **Amazon Elastic Compute Cloud** (Amazon EC2) capacity and can save you up to 90% in cost compared to On-Demand instances. By shaping your demand for the existing supply of **Amazon EC2** instance capacity, you will improve your overall resource efficiency and reduce idle capacity of the overall AWS Cloud.

5. Reduce the volume of logs you keep. By default, **CloudWatch** retains logs indefinitely. By setting limited retention time for your notebooks and training logs, you'll avoid the carbon footprint of unnecessary log storage.

6. Adopt sustainable tuning job strategy. Prefer Bayesian search over random search (and avoid grid search). Bayesian search makes intelligent guesses about the next set of parameters to pick based on the prior set of trials. It typically requires ten times fewer jobs than random search, and therefore ten times less compute resources, to find the best hyperparameters.

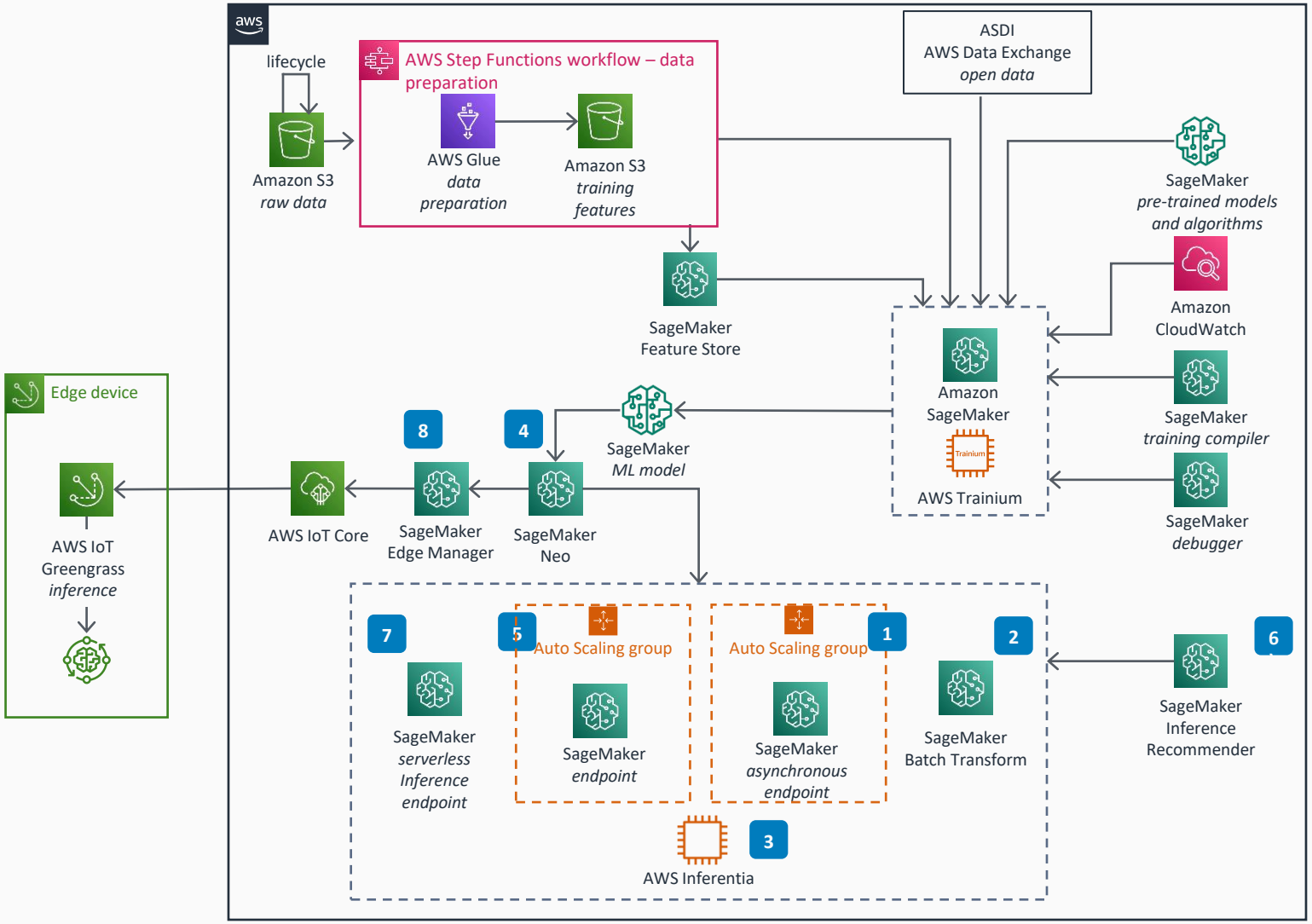
Reviewed for technical accuracy October 4, 2022

AWS Reference Architecture

Guidance for Optimizing Deep Learning Workloads for Sustainability on AWS

Inference

Use this reference architecture to reduce the environmental impact of your Deep Learning workloads.



1. If your users can tolerate some latency, deploy your model on asynchronous endpoints to reduce resources that are idle between tasks and minimize the impact of load spikes.

2. When you don't need real-time inference, use **SageMaker** batch transform. Unlike persistent endpoints, clusters are decommissioned when batch transform jobs finish.

3. **Amazon EC2** Inf1 instances (based on custom designed **AWS Inferentia** chips) have 50% higher performance per watt than g4dn.

4. Improve efficiency of your models by compiling them into optimized forms with **SageMaker Neo**.

5. Deploy multiple models behind a single endpoint. Sharing endpoint resources is more sustainable than deploying a single model behind one endpoint, and can help you cut up to 90 percent of your inference costs.

6. Right-size your endpoints by using metrics from **Amazon CloudWatch**, or by using the **Amazon SageMaker Inference Recommender**. This tool can run load testing jobs and recommend the proper instance type to host your model.

7. If your workload has intermittent or unpredictable traffic, configure autoscaling inference endpoints in **SageMaker** or use **SageMaker Serverless Inference**, which automatically launches compute resources and scales them in and out depending on traffic, which eliminates idle resources.

8. When working on Internet of Things (IoT) use cases, evaluate if ML inference at the edge can reduce the carbon footprint of your workload. When deploying ML models to edge devices, use **SageMaker Edge Manager**, which integrates with **SageMaker Neo** and **AWS IoT Greengrass**, and compress the size of models for deployment with pruning and quantization.

Reviewed for technical accuracy October 4, 2022

AWS Reference Architecture