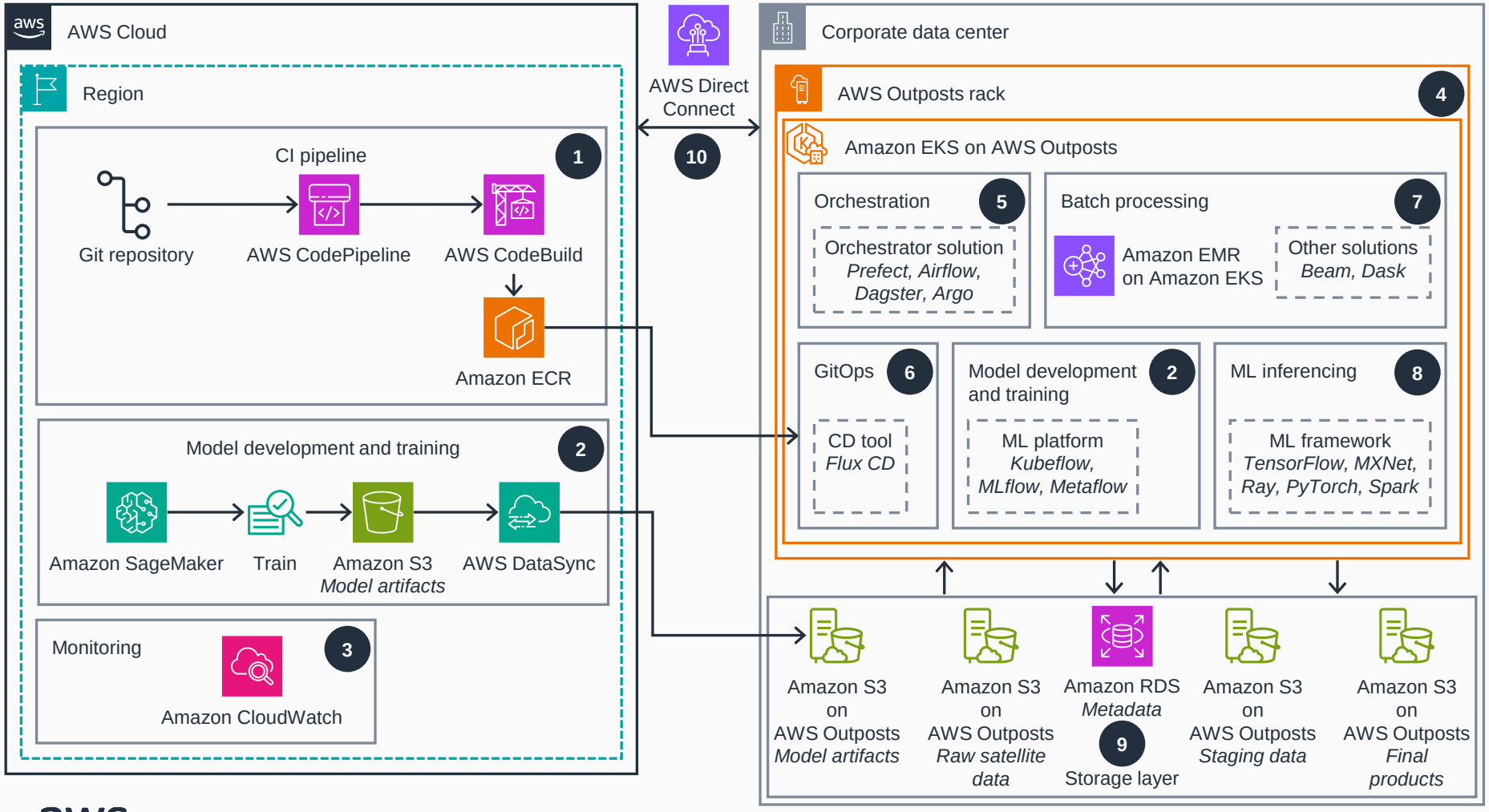
Guidance for Building Hybrid Satellite Imagery Processing Workloads on AWS

Deployment on AWS Outposts

This architecture diagram shows how to build a hybrid workload containing a satellite imagery processing pipeline deployed on an AWS Outposts rack.



AWS Reference Architecture

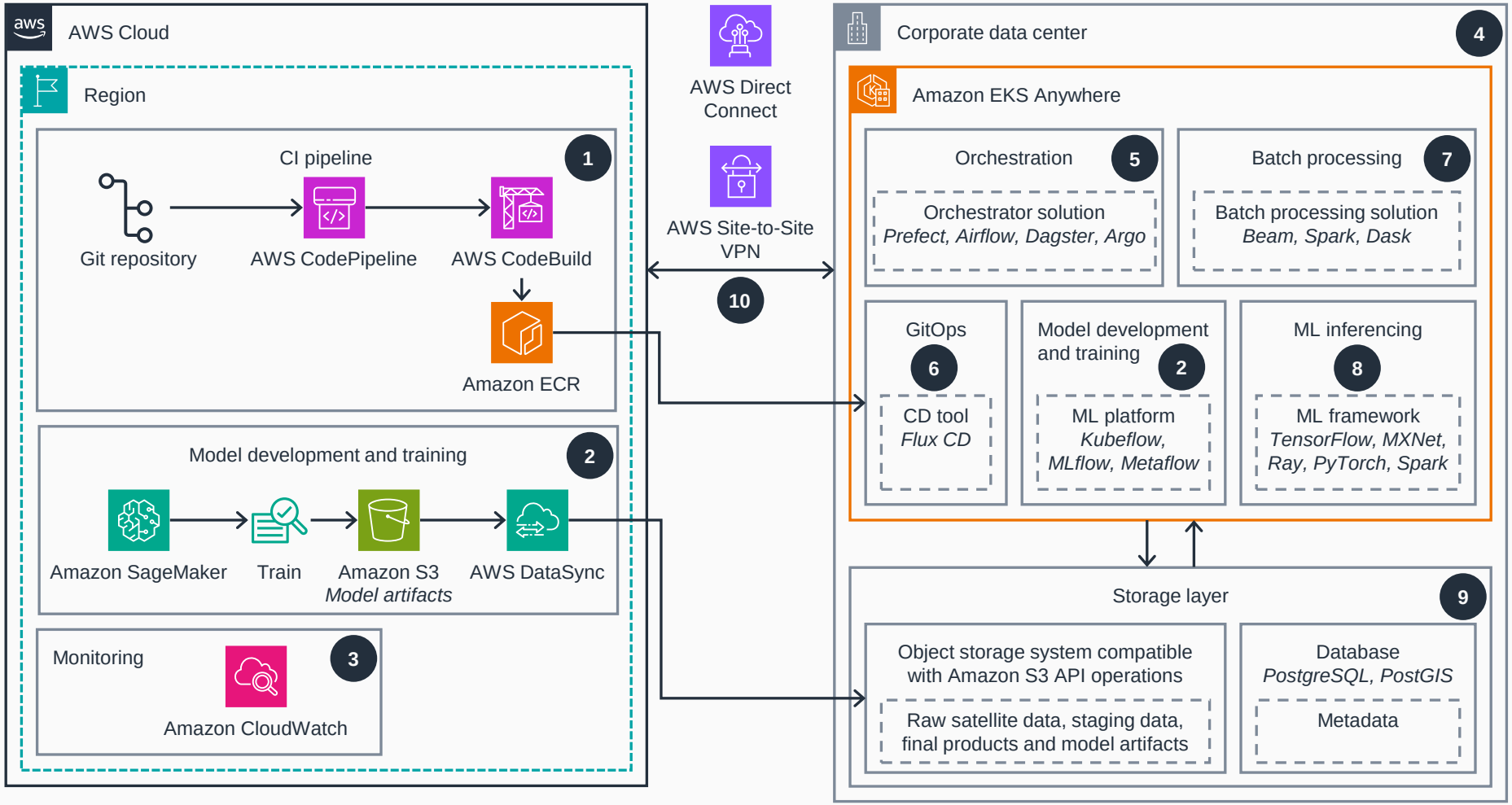
1. Create a continuous integration (CI) pipeline for your imagery processing workloads using **a** Git repository, **AWS CodePipeline**, and **AWS CodeBuild**. Store the container images in **Amazon Elastic Container Registry (Amazon ECR)**.
2. Develop and train your machine learning (ML) models using **Amazon SageMaker** in the AWS Region, an alternative ML solution in an AWS Region, or as part of the on-premises deployment. **AWS DataSync** can be used to transfer model artifacts from **Amazon Simple Storage Service (Amazon S3)** in the AWS Region to **Amazon S3 on Outposts**.
3. Use **Amazon CloudWatch** to centrally monitor AWS and on-premises resources.
4. Achieve a consistent hybrid experience and fully managed infrastructure using an **Outposts** rack for the on-premises deployment.
5. Host your processing pipeline in **Amazon Elastic Kubernetes Service (Amazon EKS)** on **Outposts**. Choose your preferred orchestrator solution, such as Prefect or Apache Airflow.
6. Following GitOps practices, use a continuous delivery (CD) tool like Flux CD to retrieve and deploy the latest container images.
7. Run batch operations to optimize processing time using **Amazon EMR** on **Amazon EKS** or another solution, such as Apache Beam.
8. Use the ML framework chosen during model development, such as TensorFlow or PyTorch, for the processing pipeline steps that require ML inferencing.
9. Store your raw and processed satellite imagery data in **Amazon S3** on **Outposts**. Maintain metadata in **Amazon Relational Database Service (Amazon RDS)**.
10. A service link will connect the **Outposts** rack with your chosen AWS Region. Optionally, you can use **AWS Direct Connect**.

Reviewed for technical accuracy January 22, 2025

Guidance for Building Hybrid Satellite Imagery Processing Workloads on AWS

Deployment on premises

This architecture diagram shows how to build a hybrid workload containing a satellite imagery processing pipeline deployed on your infrastructure.



1. Create a CI pipeline for your imagery processing workloads using a Git repository, **CodePipeline**, and **CodeBuild**. Store the container images in **Amazon ECR**.
2. Develop and train your machine learning (ML) models using **Amazon SageMaker** in the AWS Region, an alternative ML solution in an AWS Region, or as part of the on-premises deployment. **DataSync** can be leveraged to transfer model artifacts from **Amazon S3** in the AWS Region to your on-premises storage solution.
3. Use **CloudWatch** to centrally monitor AWS and on-premises resources.
4. For cases where requirements do not allow for an **Outposts** rack deployment, this hybrid architecture can be deployed directly on your infrastructure.
5. Host your processing pipeline in **Amazon EKS Anywhere**. Choose your preferred orchestrator solution, such as Prefect or Airflow.
6. Following GitOps practices, use a continuous delivery (CD) tool like Flux CD to retrieve and deploy the latest container images.
7. Run batch operations to optimize processing time using your preferred solution, such as Beam or Spark.
8. Use the ML framework chosen during model development, such as TensorFlow or PyTorch, for the processing pipeline steps that require ML inferencing.
9. Store your raw and processed satellite imagery data in your chosen object storage solution. Maintain metadata in a PostgreSQL database.
10. Connect your AWS Region deployment with your corporate data center using **AWS Site-to-Site VPN** or **Direct Connect**.

Reviewed for technical accuracy January 22, 2025

AWS Reference Architecture