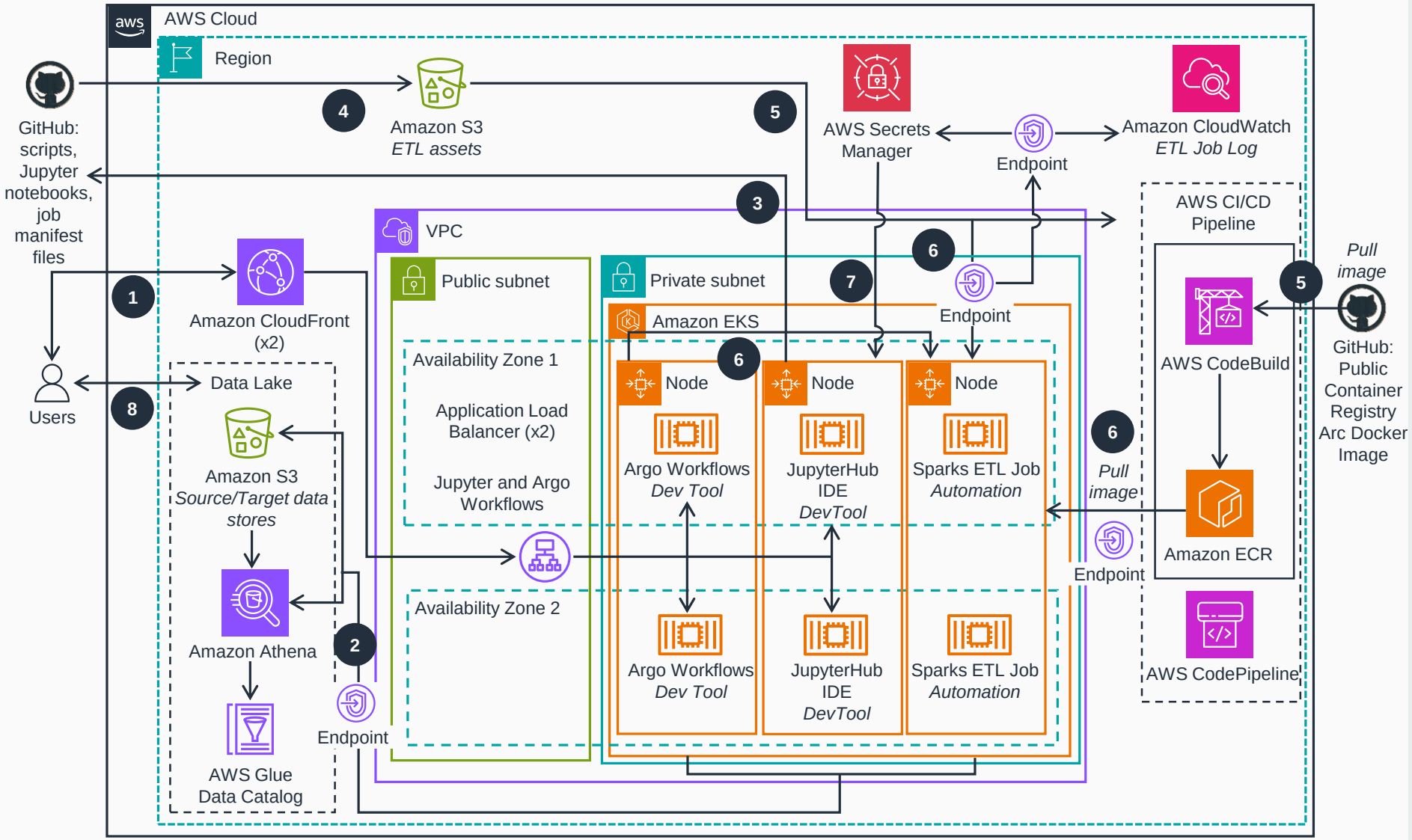
Guidance for SQL-Based ETL with Apache Spark on Amazon EKS

This architecture diagram accelerates data processing with Apache Spark on Amazon EKS. This slide shows Steps 1-4.



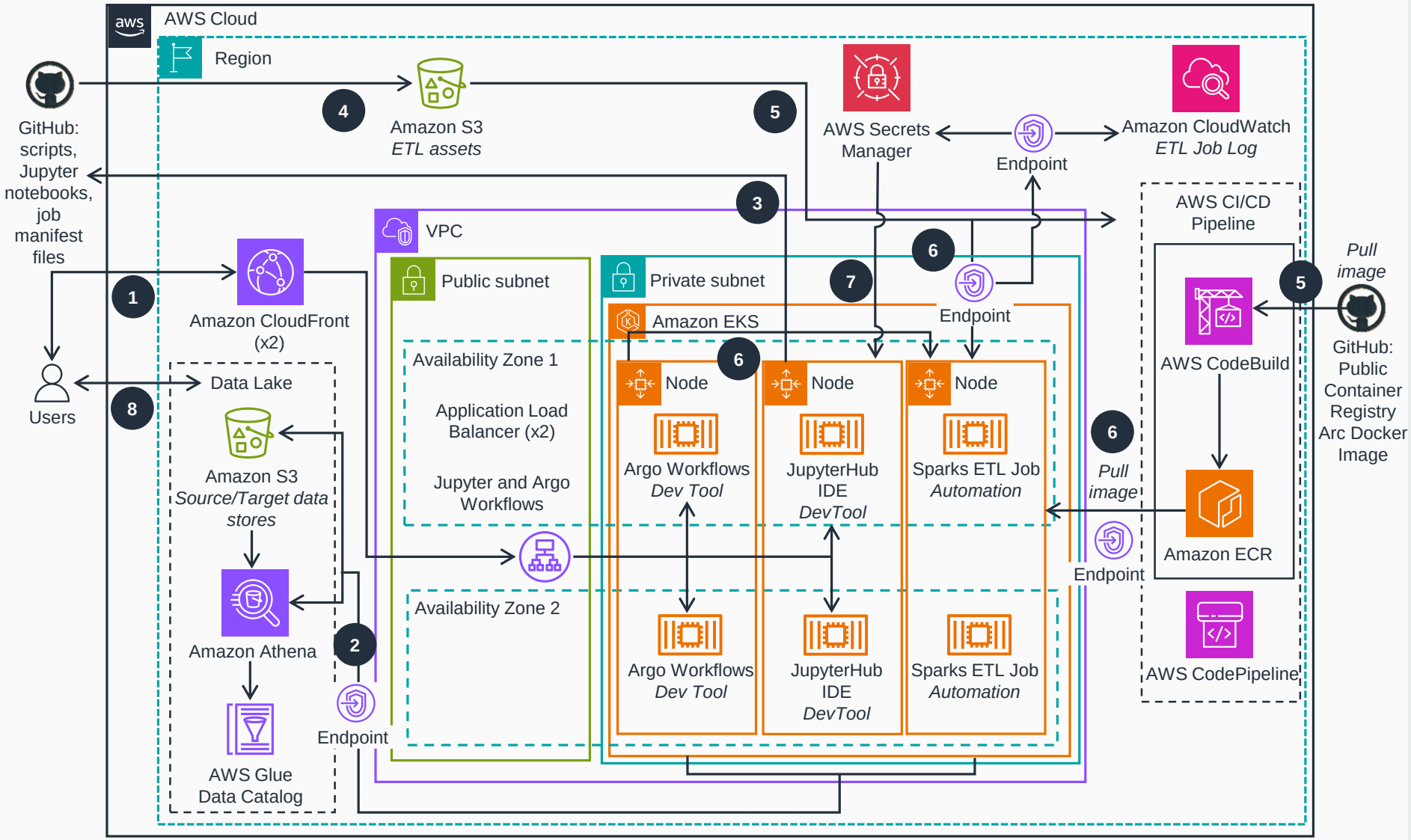
1. Interact with ETL development and orchestration tools through **Amazon CloudFront** endpoints with **Application Load Balancer** origins, which provide secure connections between clients and ETL tools' endpoints.
2. Develop, test, and schedule ETL jobs that process batch and stream data. The data traffic between ETL processes and data stores flows through **Amazon Virtual Private Cloud (Amazon VPC)** endpoints powered by **AWS PrivateLink** without leaving the AWS network.
3. JupyterHub Integrated Development Environment (IDE), Argo Workflows, and Apache Spark Operator run as containers on an **Amazon Elastic Kubernetes Service (Amazon EKS)** cluster. JupyterHub IDE can integrate with a source code repository (such as GitHub) to track ETL assets changes made by users. The assets include Jupyter notebook files and SQL scripts to be run with the Arc ETL framework.
4. Update ETL assets in the source code repository, then upload to an **Amazon Simple Storage Service (Amazon S3)** bucket. The synchronization process can be implemented by an automated continuous integration and continuous deployment (CI/CD) pipeline initiated by updates in the source code repository or performed manually.

Reviewed for technical accuracy April 8, 2024

AWS Reference Architecture

Guidance for SQL-Based ETL with Apache Spark on Amazon EKS

This architecture diagram accelerates data processing with Apache Spark on Amazon EKS. This slide shows Steps 5-8.



5 You can optionally change Docker build source code uploaded from a code repository to the **S3** ETL asset bucket. It activates an **AWS CodeBuild** and **AWS CodePipeline** CI/CD pipeline to automatically rebuild and push the Arc ETL Framework container image to an **Amazon Elastic Container Registry (Amazon ECR)** private registry.

6 Schedule ETL jobs through Argo Workflows to run on an **Amazon EKS** cluster. These jobs automatically pull the Arc container image from **Amazon ECR**, download ETL assets from the artifact **S3** bucket, and send application logs to **Amazon CloudWatch**. **VPC** endpoints secure access to all AWS services.

7 As an authenticated user, you can interactively develop and test notebooks as ETL jobs in JupyterHub IDE, which automatically retrieves log-in credentials from **AWS Secrets Manager** to validate sign-in user requests.

8 Access the ETL output data stored in the **S3** bucket that supports the transactional data lake format. You can query the Delta Lake tables through **Amazon Athena** integrated with **AWS Glue Data Catalog**.

Reviewed for technical accuracy April 8, 2024

AWS Reference Architecture