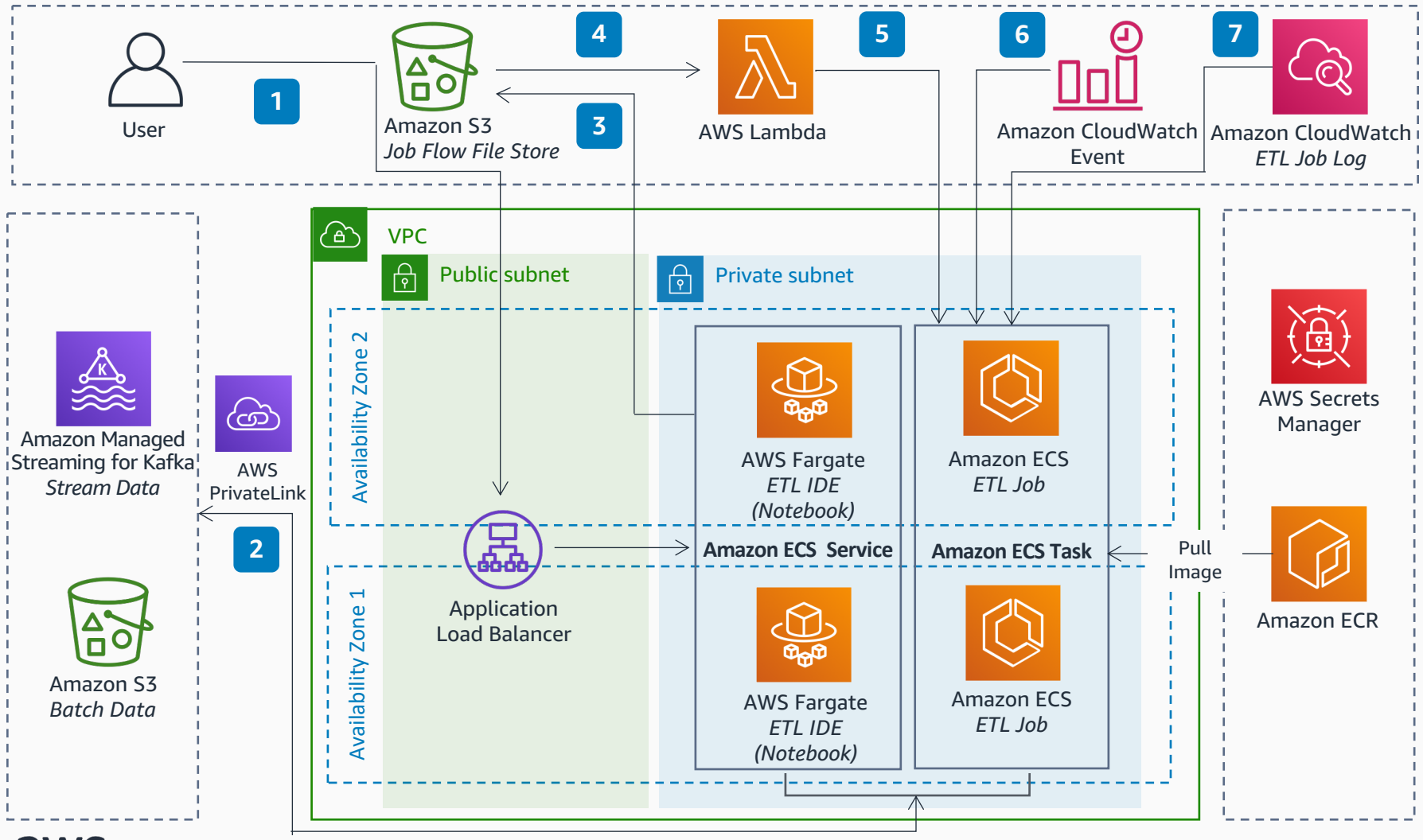
SQL Based Data Processing in Amazon ECS

Build a configuration-driven, codeless extract-transform-load (ETL) alternative using a containerized ETL framework (ARC) that simplifies and accelerates data processing with Apache Spark.



1. User creates an extract-transform-load (ETL) data pipeline based on ARC framework and SQL scripts in an interactive ARC Jupyter Notebook. The pipeline is hosted in **Amazon Elastic Container Service (Amazon ECS)**.
2. The Notebook and ETL jobs process batch and stream data via **AWS PrivateLink**. The traffic between ETL processes and data stores does not leave the Amazon network.
3. ARC Jupyter notebook produces a job flow configuration JSON file; user uploads the file and SQL scripts to **Amazon S3** via CI/CD automated deployment process or manually.
4. An **Amazon S3** file arrival event triggers an **AWS Lambda** function.
5. The Lambda function spins up an **Amazon ECS** task to process batch data in a transient way, or to process stream data continuously in a long-running container. Each job has isolated compute resources.
6. **Amazon CloudWatch Events** schedules and orchestrates regular ARC ETL jobs and ECS tasks with **AWS Fargate** or **Amazon EC2** launch types.
7. ARC ETL job generates application logs for each data process stages, at a granular level. **Amazon CloudWatch** offers monitoring and alerting capabilities.

AWS Reference Architecture