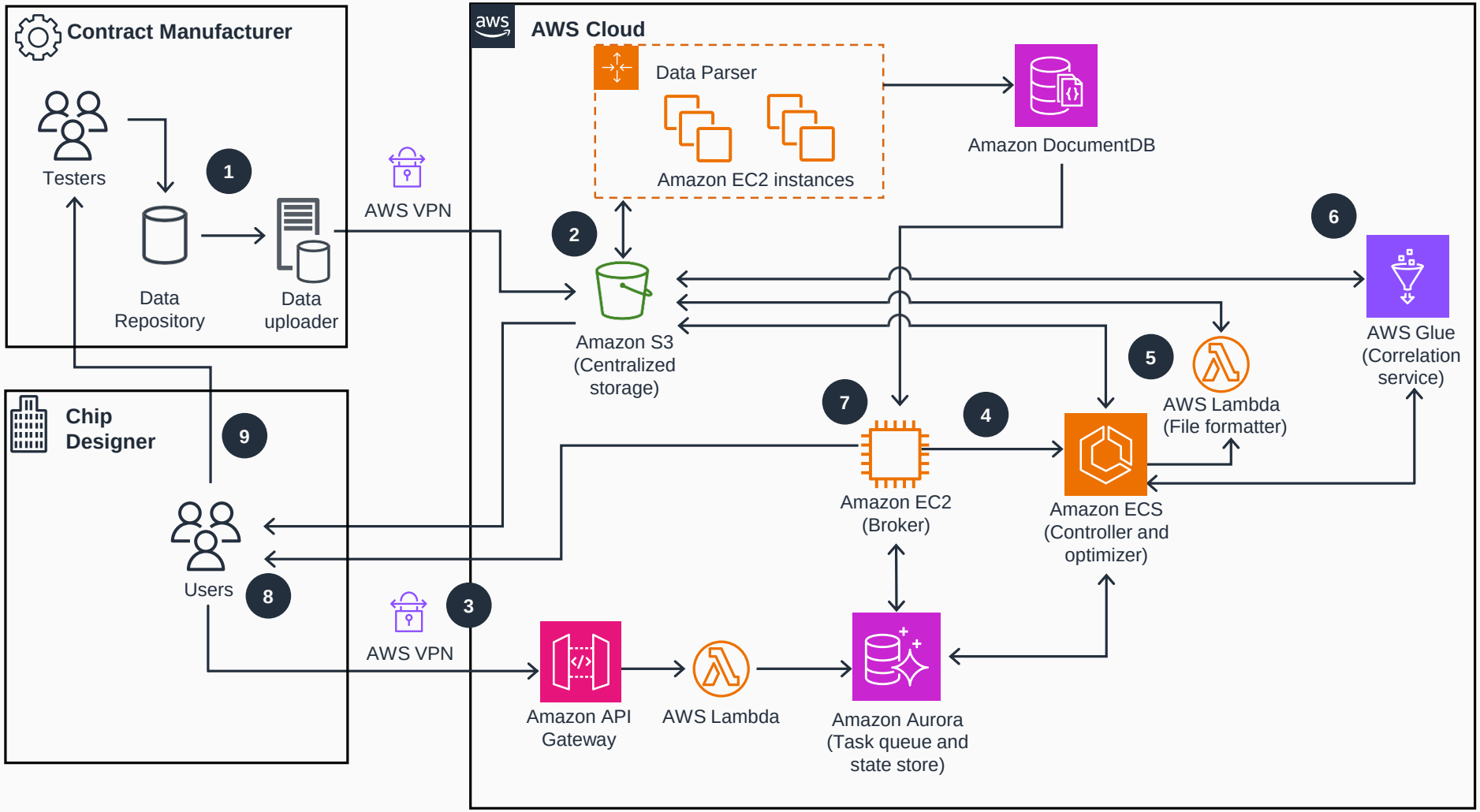
Guidance for Reducing Workbench Time and Cost on AWS

This architecture diagram shows how to run workbench test results and parameters on AWS so chip designers and contract manufactures can run test jobs and quickly analyze test results, saving both time and money.



1. A tester stages data on the data repository and uses the data uploader to upload data to an **Amazon Simple Storage Service (Amazon S3)** bucket for centralized storage.
2. **Amazon Elastic Compute Cloud (Amazon EC2)** instances along with **Amazon EC2 Auto Scaling** parse data into test files. Data is stored in the **S3** bucket and a database in **Amazon DocumentDB**.
3. A user submits a Dynamic Parameter Reduction (DPR) request through **Amazon API Gateway**. An **AWS Lambda** function stores the request in the task queue and state store in an **Amazon Aurora** database.
4. The broker (**EC2** instance) pulls the DPR request from the task queue, queries data from **Amazon DocumentDB**, updates the requested input state store, and invokes the controller.
5. The controller, running on **Amazon Elastic Container Service (Amazon ECS)**, initiates **Lambda**, which is running the file formatter, to prepare data in the **S3** bucket. The file formatter updates the status in the **Aurora** state store.
6. The controller also initiates the correlation service running on **AWS Glue**, and the results are sent to the Optimizer (**Amazon ECS**). The **S3** bucket stores optimized results.
7. The broker is notified that the DPR job completed from state store, and the user receives a notification email.
8. User(s) collect results from the **S3** bucket.
9. After the results are reviewed, the user adjusts parameters and provides these new parameters to testers to drive future testing.

Reviewed for technical accuracy January 8, 2024

AWS Reference Architecture