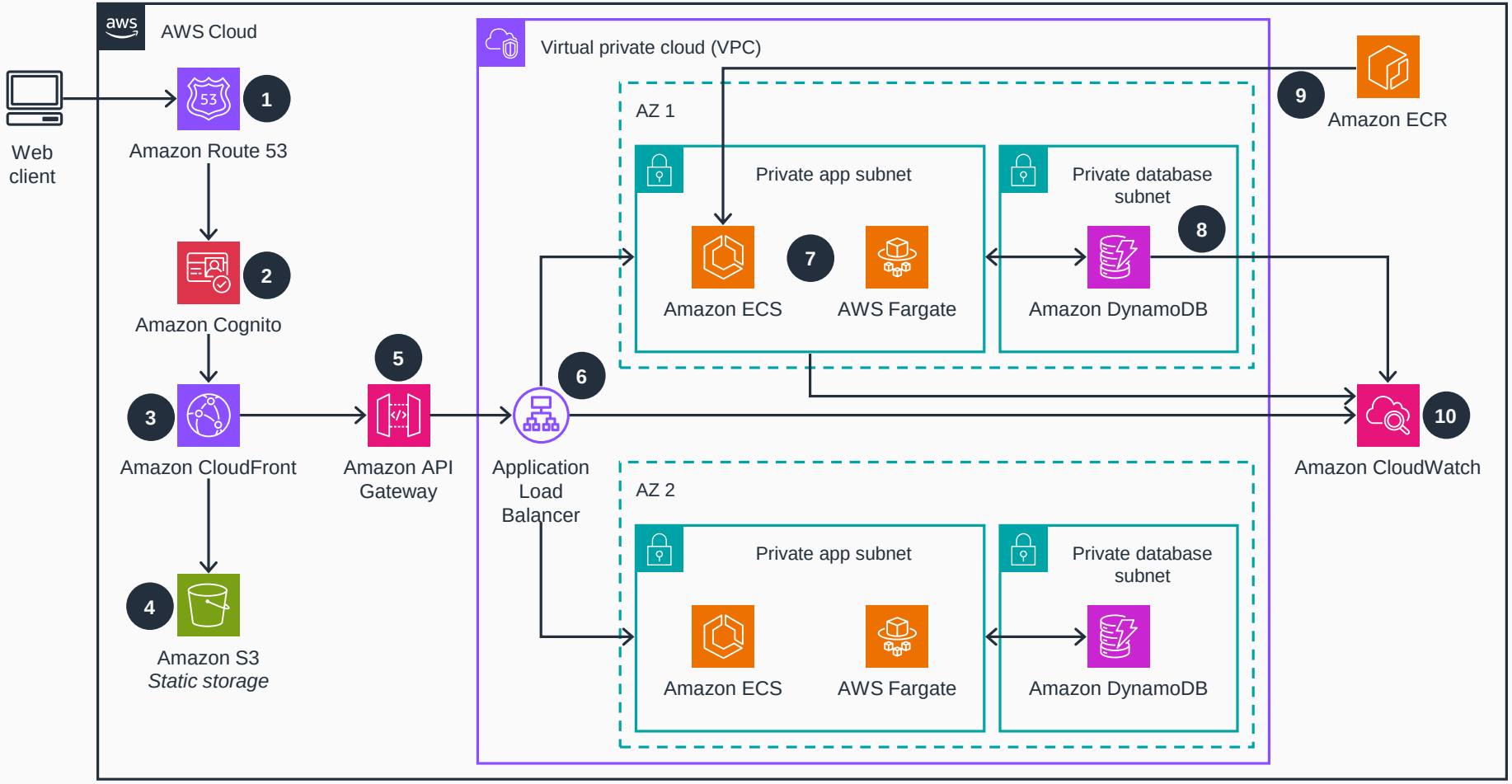
Guidance for Building a Containerized and Scalable Web Application on AWS

This architecture diagram shows how you can deploy a scalable, secure three-tier web application using containerization on AWS without needing extensive knowledge in containerization or infrastructure management.



1. Route traffic from your web client based on the request path for static and dynamic content using domain name service (DNS) **Amazon Route 53**.
2. Protect and control access to your web application using **Amazon Cognito**.
3. Use a content delivery network (CDN) like **Amazon CloudFront** to reduce the latency for delivering your static content.
4. Use **Amazon Simple Storage Service (Amazon S3)** to store static content and backups.
5. Handle all incoming API calls and traffic management with authorization, access control, and throttling using **Amazon API Gateway**.
6. Configure **Application Load Balancer** to be internet-facing, and use it to distribute web traffic to your application across multiple Availability Zones (AZs).
7. Run the application on **Amazon Elastic Container Service (Amazon ECS)**, and use **AWS Fargate** for serverless compute. Send API calls for dynamic content.
8. Retrieve application data and content from **Amazon DynamoDB** anytime there is an API call.
9. Store the container image running the application in **Amazon Elastic Container Registry (Amazon ECR)**. Use **Amazon ECS** to pull the image to run the application.
10. Use **Amazon CloudWatch** to monitor and observe the application and all the resources.

Reviewed for technical accuracy September 18, 2023

AWS Reference Architecture