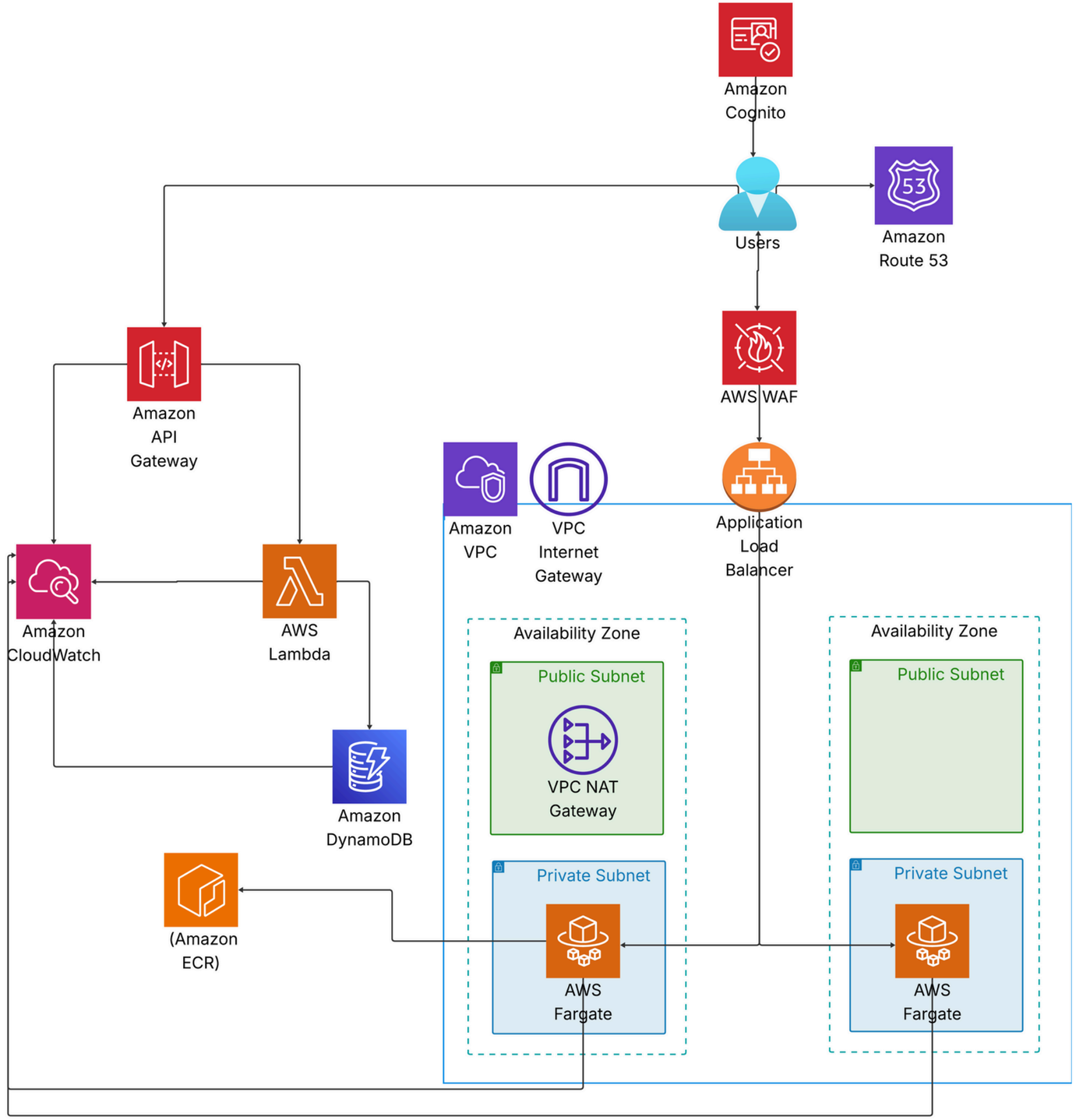
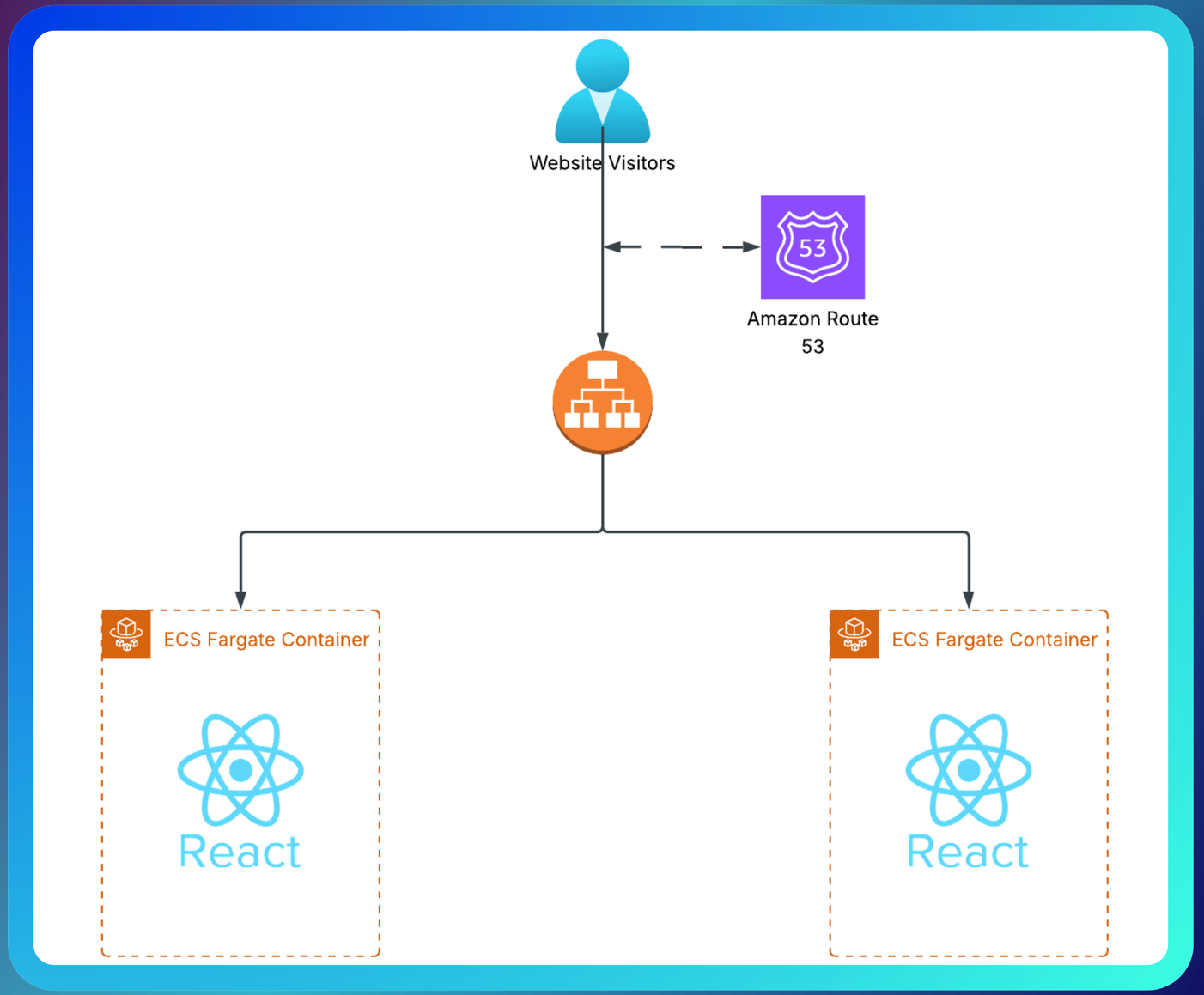
Levi9 Technology Services, Internship task 2025.

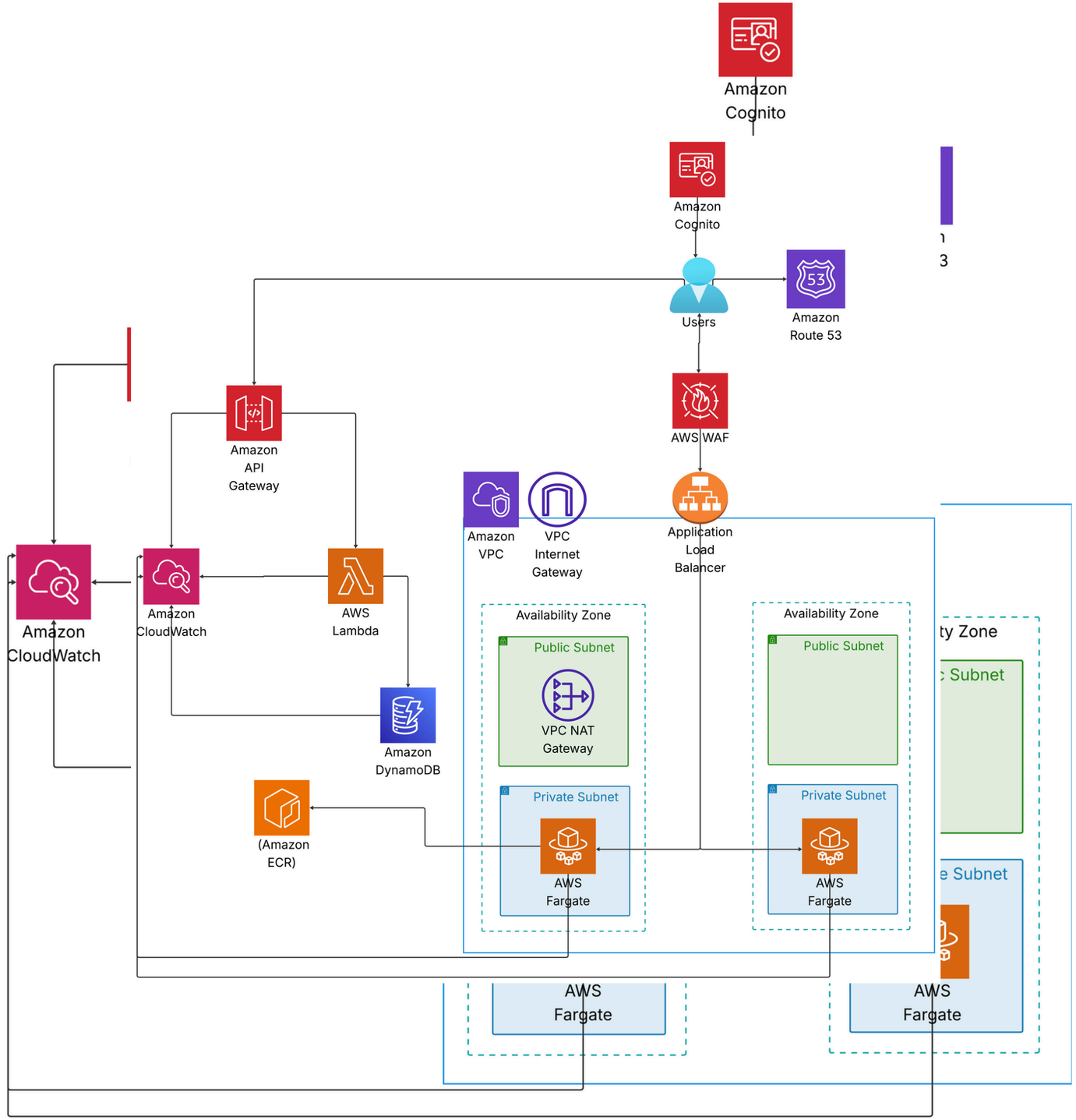
AUTOMATED DEPLOYMENT OF A FULL-STACK SERVERLESS APPLICATION


Amazon
Cognito
Users
Amazon
Route 53
Amazon
API
Gateway
AWS WAF
Amazon
VPC
VPC
Internet
Gateway
Application
Load
Balancer
Amazon
CloudWatch
AWS
Lambda
Availability Zone
Public Subnet
VPC NAT
Gateway
Availability Zone
Public Subnet
Amazon
DynamoDB
Private Subnet
Private Subnet
(Amazon
ECR)
AWS
Fargate
AWS
Fargate

ECS

- AWS ECS (Elastic Container Service) is a fully managed container orchestration service
- ECS supports two launch types: **EC2** and **Fargate**
- Our application runs on Fargate containers
- Docker image is pulled from private **ECR** repository




Amazon
Cognito
Amazon
Cognito
Users
Amazon
Route 53
AWS WAF
Amazon
API
Gateway
Amazon
VPC
VPC
Internet
Gateway
Application
Load
Balancer
Amazon
CloudWatch
Amazon
CloudWatch
AWS
Lambda
Availability Zone
Public Subnet
VPC NAT
Gateway
Private Subnet
AWS
Fargate
Availability Zone
Public Subnet
Private Subnet
AWS
Fargate
Amazon
DynamoDB
(Amazon
ECR)
AWS
Fargate
AWS
Fargate

DOCKERFILE ENTRYPOINT SCRIPT

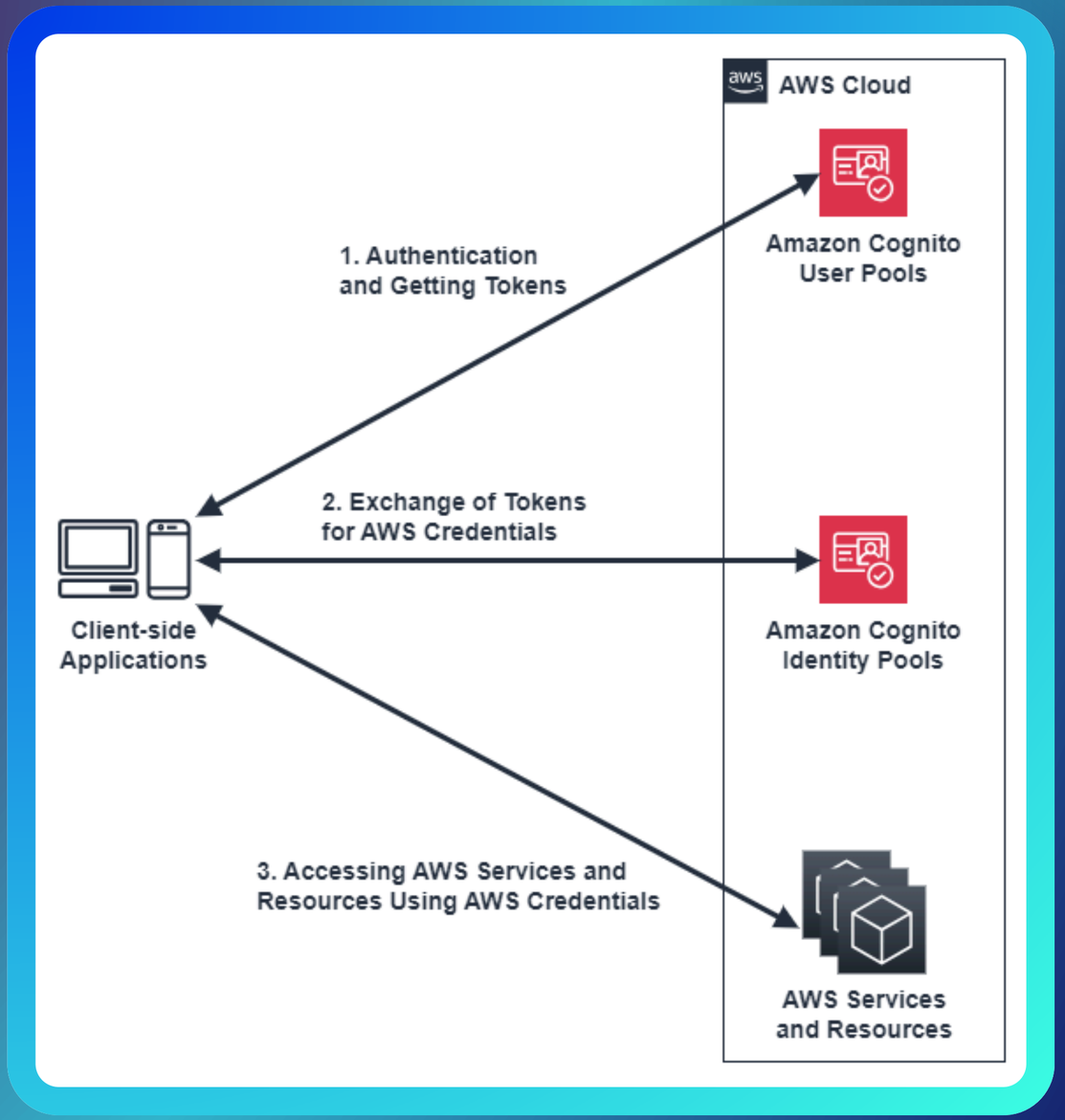
```
#!/bin/sh

JS_DIR="/usr/share/nginx/html/static/js"

API_URL_STRING="https://fjngnfrqig.execute-api.us-east-1.amazonaws.com/prod"
USER_POOL_ID_STRING="eu-west-1_oCmXg40Hj"
APP_CLIENT_ID_STRING="1lbtdpak5v3mloh3143tes7ivd"

API_URL_REPLACEMENT=${API_URL_REPLACEMENT:-$API_URL_STRING}
USER_POOL_ID_REPLACEMENT=${USER_POOL_ID_REPLACEMENT:-$USER_POOL_ID_STRING}
APP_CLIENT_ID_REPLACEMENT=${APP_CLIENT_ID_REPLACEMENT:-$APP_CLIENT_ID_STRING}

for file in $JS_DIR/*.js; do
    echo "Replacing URL in $file..."

    sed -i "s|$API_URL_STRING|$API_URL_REPLACEMENT|g" "$file"
    sed -i "s|$USER_POOL_ID_STRING|$USER_POOL_ID_REPLACEMENT|g" "$file"
    sed -i "s|$APP_CLIENT_ID_STRING|$APP_CLIENT_ID_REPLACEMENT|g" "$file"
done

echo "Starting Nginx..."
exec "$@"
```

COGNITO

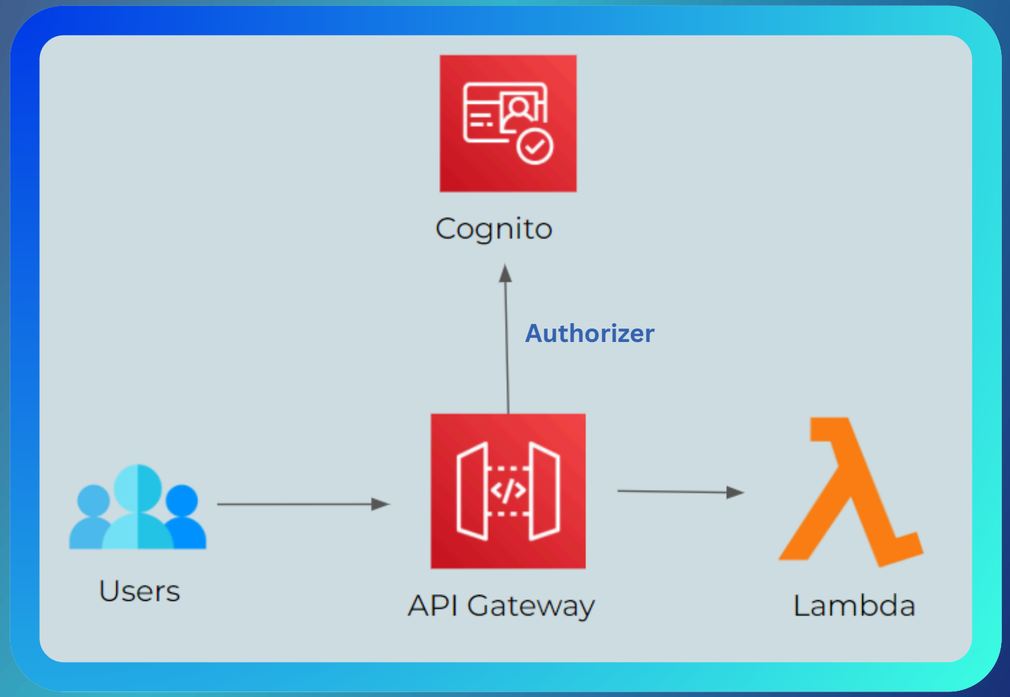
- Fully managed identity service for web and mobile applications

- User pool - User directory to manage user registration and authentication
- Supports customizable sign-up/sign-in workflows

- Applications must connect through App Clients (trusted application in Cognito)



COGNITO AUTHORIZER



API GATEWAY

API Gateway routes client HTTP requests to AWS services like Lambda, acting as a secure, scalable front door for applications.

Lambda Integration allows API Gateway to forward authenticated requests to serverless backend functions for processing.

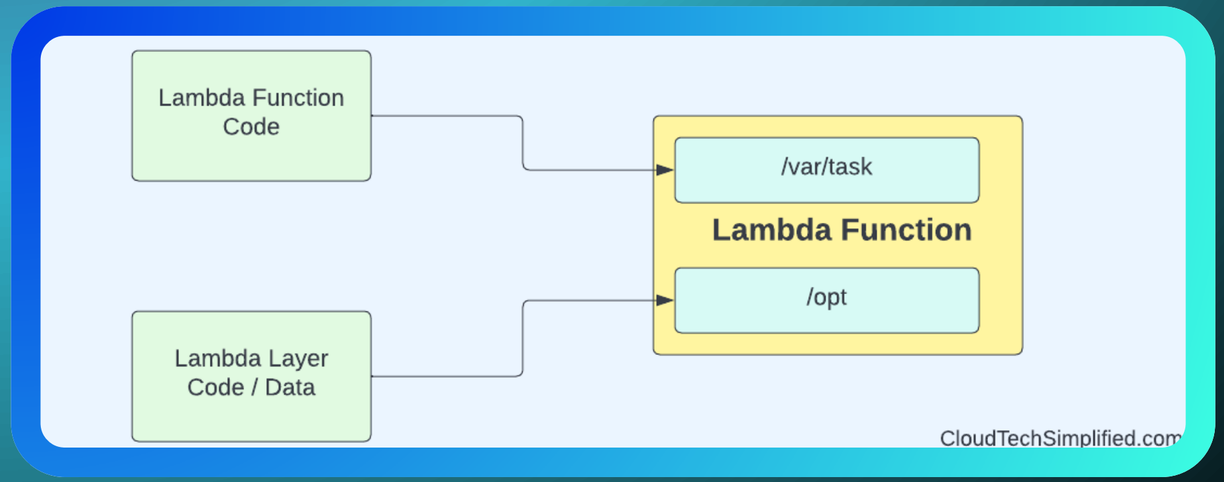
IAM ROLES

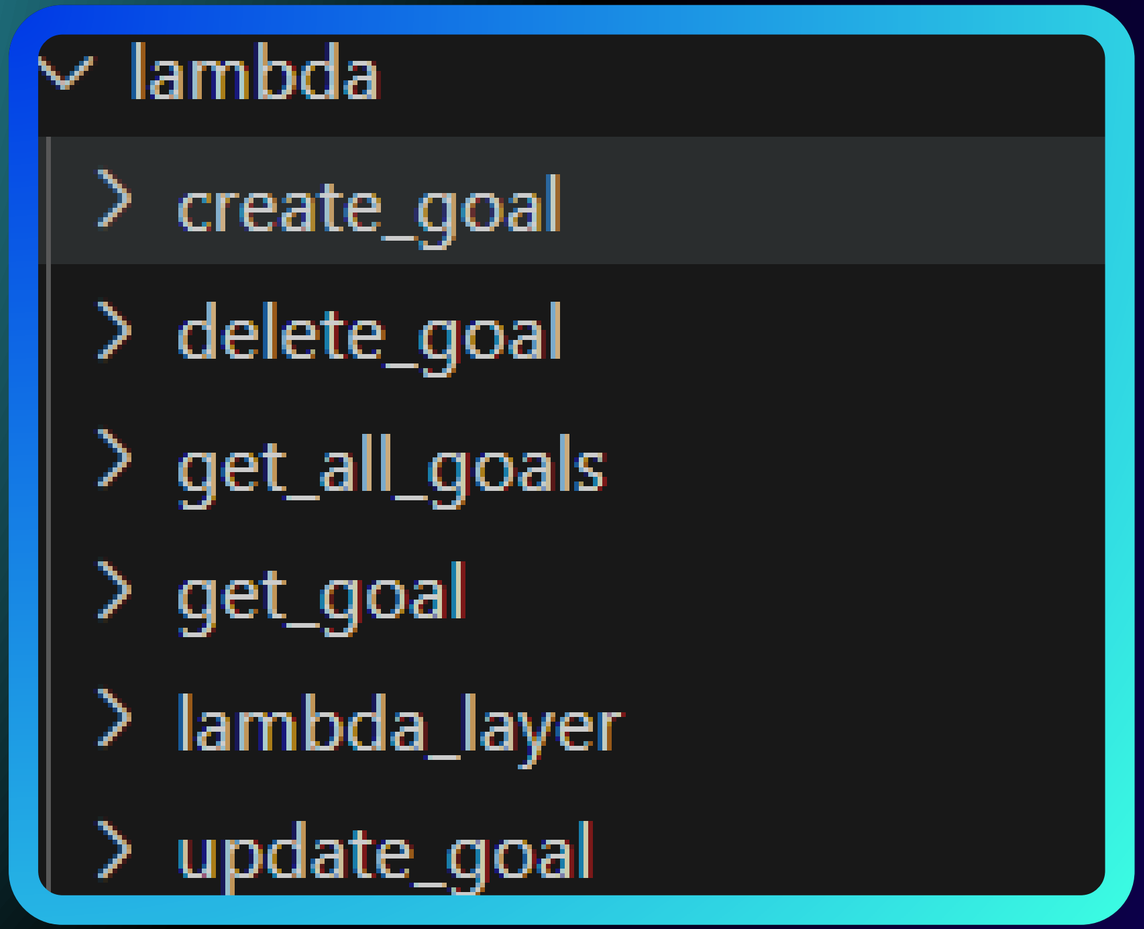
IAM roles securely grant AWS services the exact access they need without hardcoding credentials.

Gave Lambda permissions to access DynamoDB, and Fargate permissions to pull images from ECR.

LAMBDA FUNCTIONS

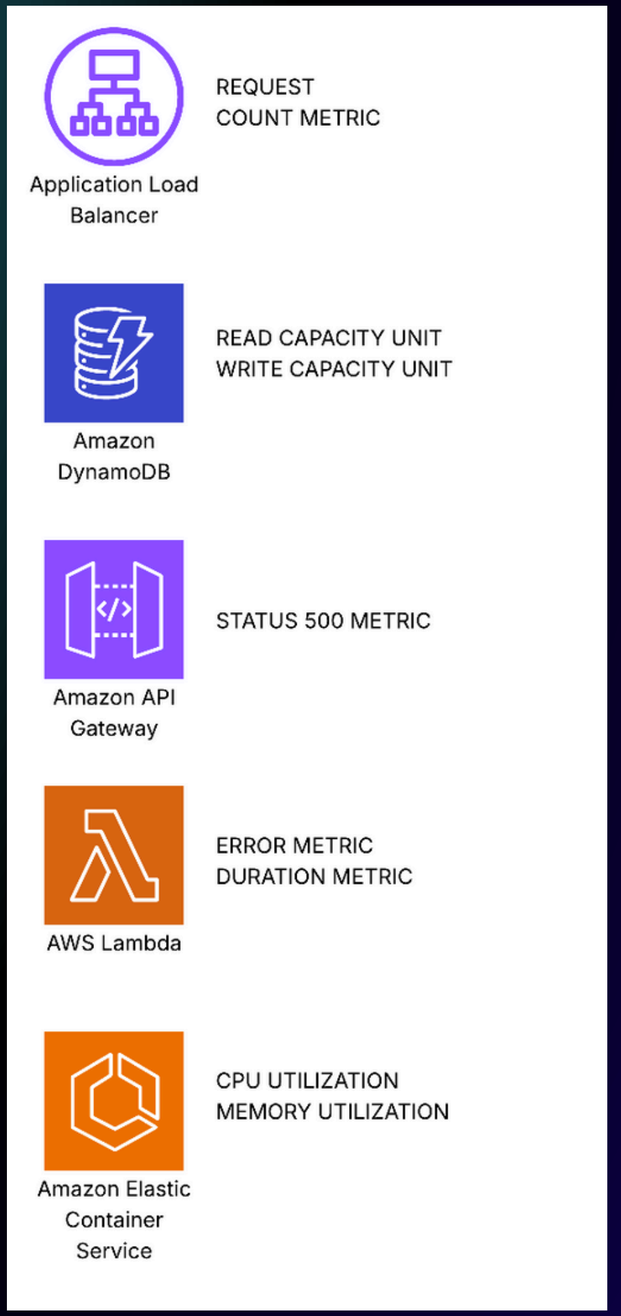
AWS Lambda: Serverless service to run code on demand, automatically scaling with no server management.





MONITORING CLOUDWATCH

Monitoring is important because it enables the tracking of progress, identification of issues, and ensures that goals are being met effectively and efficiently.



AWS CODEPIPELINE