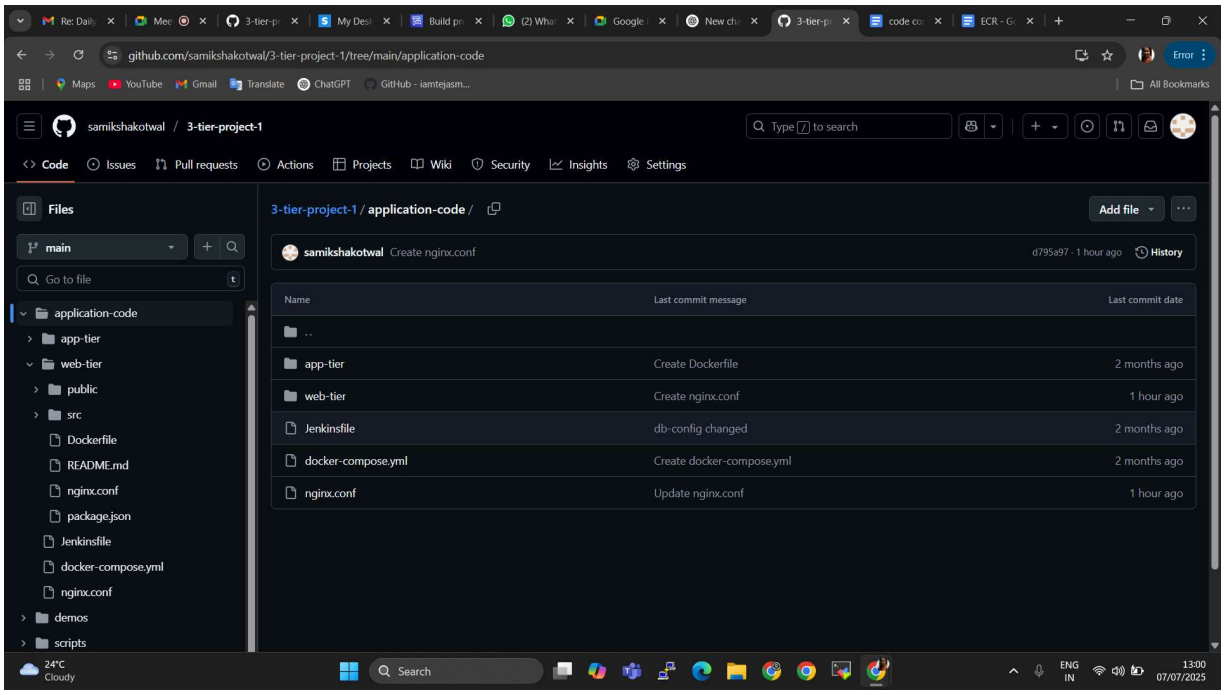
Code Commit (build project)

1. Create Project Files

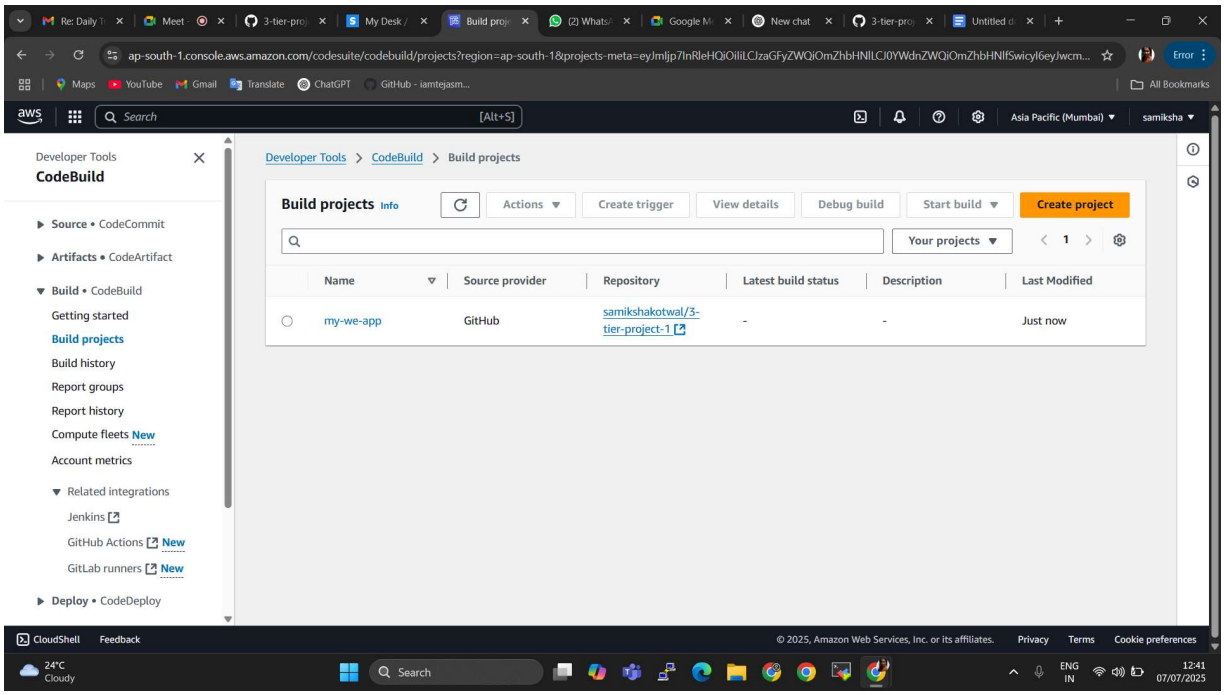
- https://github.com/samikshakotwal/3-tier-project-1.git



2. Create a CodeBuild Project

- Go to AWS Console > CodeBuild > Create Build Project.
- Project Name: my-web-app
- Source Provider: AWS CodeCommit
- Repository: Choose your repo
  - Choose: GitHub
  - Click "Connect to GitHub"
- (Login and authorize AWS)
- Environment:
  - OS: Amazon Linux 2
  - Runtime: Choose your runtime (e.g., Node.js)
  - Privileged: Checked if using Docker

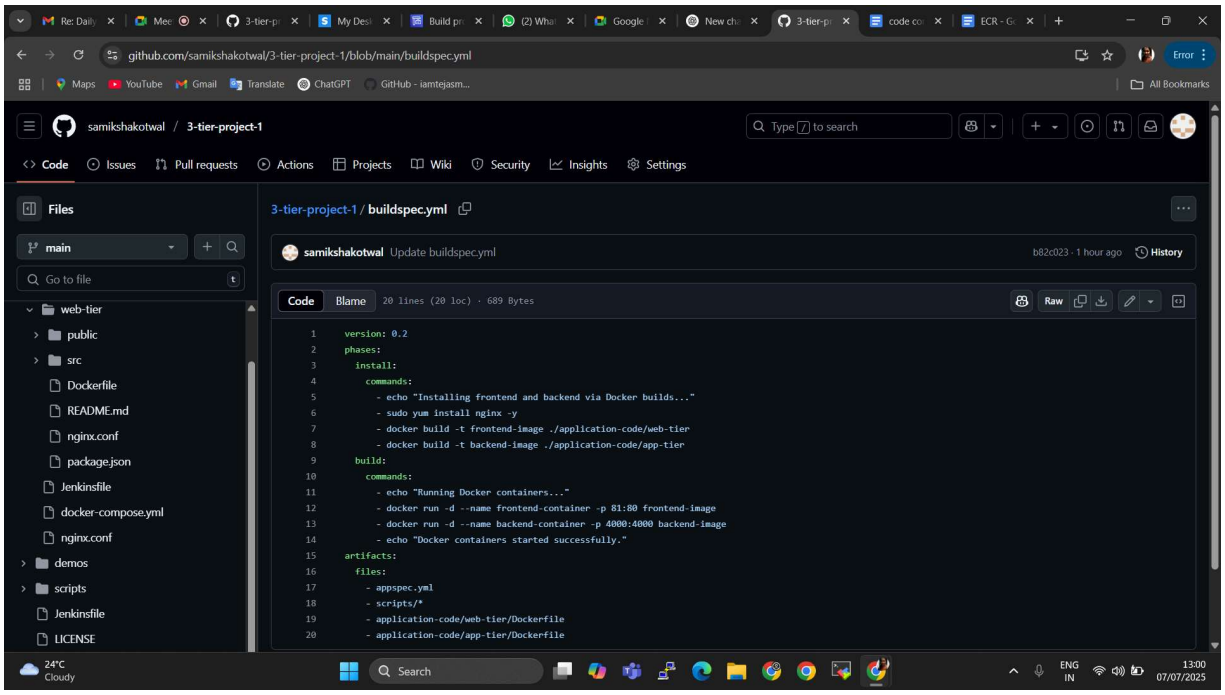
    - Service Role: Create new or choose existing IAM role
- Buildspec: Use buildspec.yml from source
- Artifacts: No artifacts or S3 (optional)
- Click “Create Build Project”.



3. Create buildspec.yml in Github

```
version: 0.2
phases:
  install:
    commands:
      - echo "Installing frontend and backend via Docker builds..."
      - sudo yum install nginx -y
      - docker build -t frontend-image ./application-code/web-tier
      - docker build -t backend-image ./application-code/app-tier
  build:
    commands:
      - echo "Running Docker containers..."
      - docker run -d --name frontend-container -p 81:80 frontend-image
      - docker run -d --name backend-container -p 4000:4000 backend-image
      - echo "Docker containers started successfully."
artifacts:
```

```
files:
  - appspec.yml
  - scripts/*
  - application-code/web-tier/Dockerfile
  - application-code/app-tier/Dockerfile
```

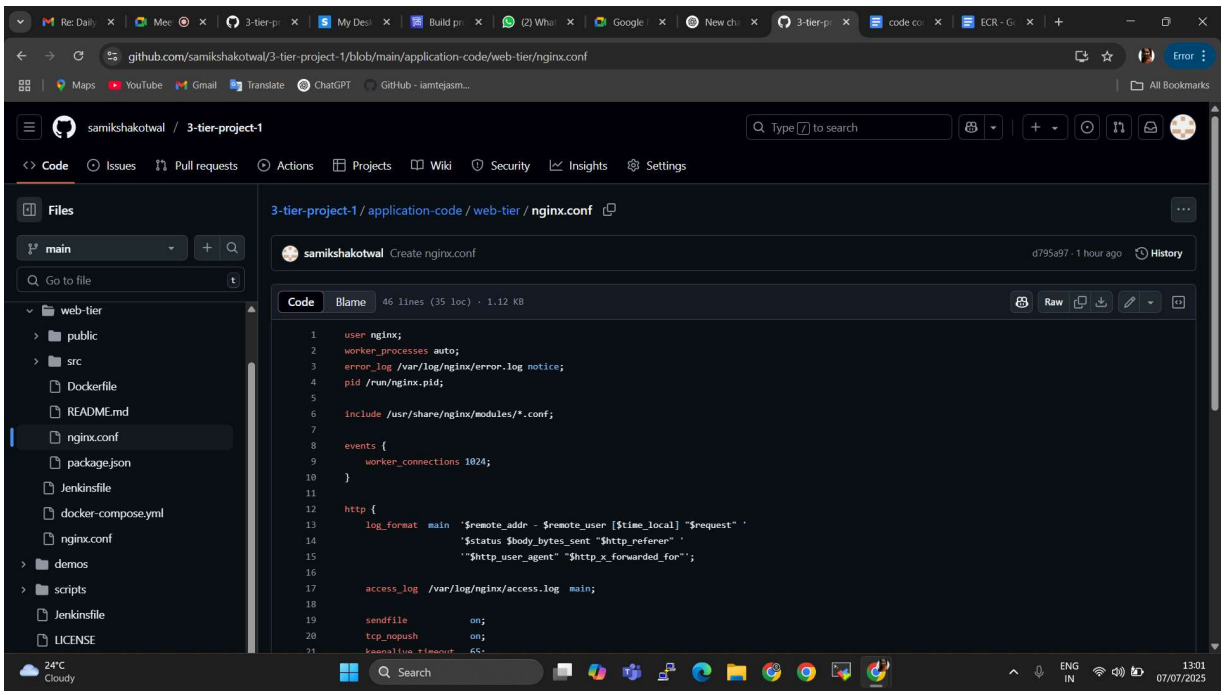


4. **Go to application-code/web-tier/**

   - Create Dockerfile

```
# Stage 1: Build the React app
FROM public.ecr.aws/docker/library/node:18 AS builder
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN  npm run build
# Stage 2: Serve with NGINX
FROM nginx:stable-alpine
# Remove default config
RUN rm /etc/nginx/nginx.conf
```

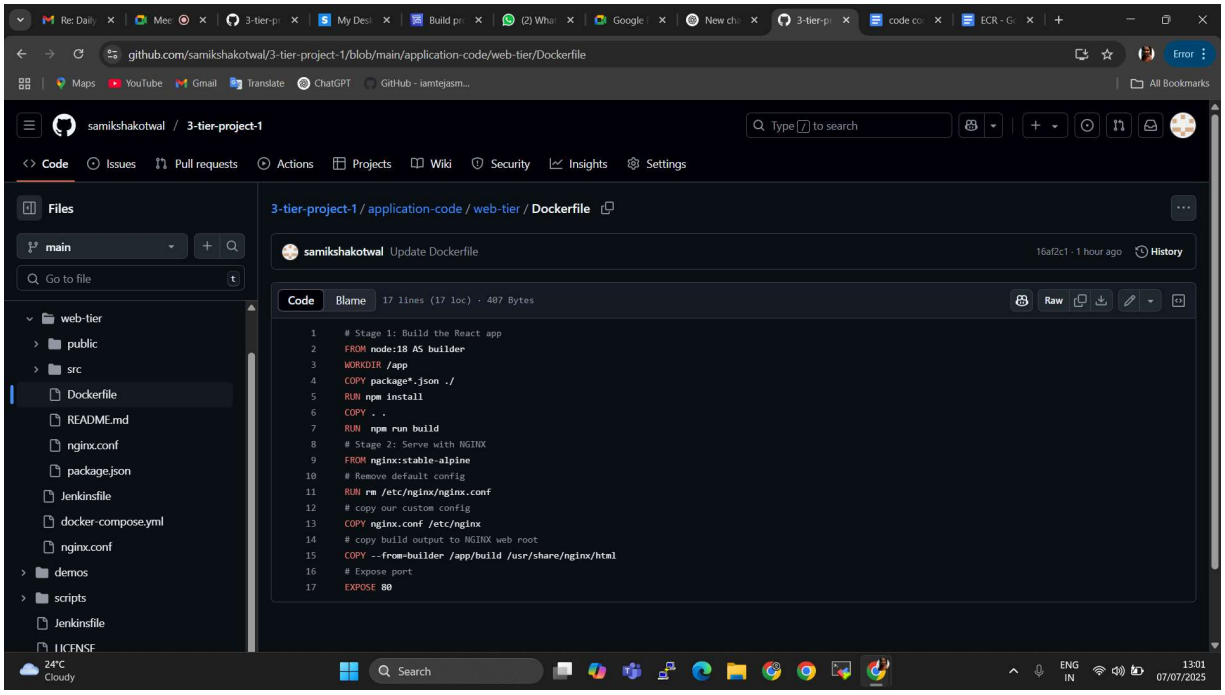
```
# copy our custom config
COPY nginx.conf /etc/nginx
# copy build output to NGINX web root
COPY --from=builder /app/build /usr/share/nginx/html
# Expose port
EXPOSE 80
```



- Create nginx.conf

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
```

```
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    include /etc/nginx/conf.d/*.conf;

    server {
        listen 80;
        server_name localhost;

        location / {
            proxy_pass http://localhost:81;   # Frontend
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
        }

        location /api/ {
            proxy_pass http://localhost:4000/;   # Backend
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
        }
    }
}
```



5. Configure Project

- Then click "Start build"
- Monitor build logs to see progress