ShopIT Application Deployment - Challenges & Solutions

Deployment Challenges & Solutions

Project: ShopIT - eCommerce Application

Description for Resume:

- Developed and deployed a full-stack eCommerce application using Redux Toolkit, React.js, Node.js, and Express.js.
- Successfully deployed the frontend and backend on AWS EC2 instances with NGINX and PM2 for process management.
- Integrated MongoDB Atlas as the cloud database and ensured secure API communications.

Challenges Faced (for Interview Discussion):

1. CORS Issues:
   - Problem: Frontend (hosted on CloudFront) tried to call backend APIs on EC2 IP, triggering CORS errors in the browser.
   - Resolution: Either backend had to explicitly allow frontend domain with CORS headers, or frontend had to be served from the same EC2 server to avoid cross-origin.

2. HTTPS Restriction with CloudFront:
   - Problem: CloudFront enforces HTTPS, but backend was hosted on HTTP (EC2 IP).
   - Resolution: Shifted frontend hosting from CloudFront to EC2 to avoid mixed content issues.

3. NGINX Configuration:
   - Learned to configure NGINX as a reverse proxy to route API calls to backend and serve static frontend files.

4. File Transfer & Permissions:
   - Faced permission denied errors when uploading React build to EC2 due to wrong ownership on

ShopIT Application Deployment - Challenges & Solutions

/var/www/html.

- Used sudo and proper permission settings to resolve.

5. MongoDB Connection:

- Used MongoDB Compass initially, then connected to MongoDB Atlas cluster in backend securely via connection URI.

6. Domain Restrictions in CloudFront:

- Faced error when setting EC2 IP as origin in CloudFront because IPs are not allowed.
- Understood CloudFront needs domain or S3, not public IPs.

7. Final Decision:

- Moved both frontend and backend to EC2.
- Ensured single origin setup (same IP for frontend + backend), eliminating CORS and HTTPS mismatch.

Result:

- Application is accessible via single EC2 public IP without CORS or HTTPS errors.
- Gained deep understanding of cloud deployment, NGINX proxying, static vs dynamic content handling, and cross-origin policies.