🚀 Push Changes and Redeploy - Step by Step

✅ What I Fixed

I created two new files at your **repository root** to fix the Cloud Build error:

1. **`Dockerfile`** - Tells Cloud Build how to build your app from the `nextjs_space/` subdirectory
2. **`.dockerignore`** - Tells Docker which files to exclude from the build

Key improvements in the new Dockerfile:

- ✅ Correctly references `nextjs_space/` subdirectory
- ✅ Includes a build-time `DATABASE_URL` argument (with default placeholder)
- ✅ Generates Prisma client during build
- ✅ Enables Next.js standalone output mode
- ✅ Creates the uploads directory (ready for GCS mounting)

Both files are committed to your local Git repository with commit hash: `fba711a`

📤 Step 1: Push to GitHub

Your changes are committed locally but need to be pushed to GitHub so Cloud Run can access them.

Option A: Push via Command Line (If you have Git configured)

```
cd /home/ubuntu/hotel_shift_log
git push origin master
```

If this works, skip to Step 2!

Option B: Push via GitHub Desktop or Git GUI

If you're using GitHub Desktop or another Git GUI:

1. Open your Git client
2. Navigate to the `Hotel_Front_Desk_Shift_Log` repository
3. You should see the commit: "Add Dockerfile and .dockerignore at repository root for Cloud Run deployment"
4. Click **Push** or **Sync**

Option C: Set up Remote and Push (If remote not configured)

```
cd /home/ubuntu/hotel_shift_log

# Add your GitHub repository as remote (replace with YOUR repo URL)
git remote add origin https://github.com/jpress-knighted/
Hotel_Front_Desk_Shift_Log.git

# Or if you use SSH:
# git remote add origin git@github.com:jpress-knighted/Hotel_Front_Desk_Shift_Log.git

# Push to GitHub
git push -u origin master
```

Replace the URL with your actual GitHub repository URL!

Verify Push Succeeded

1. Go to your GitHub repository: https://github.com/jpress-knighted/Hotel_Front_Desk_Shift_Log
2. Check that you can see:
   - `Dockerfile` in the root directory
   - `.dockerignore` in the root directory
3. ✅ If you see these files, push successful!

🔄 Step 2: Trigger New Deployment in Cloud Run

Now that the Dockerfile is in your GitHub repo, trigger a new build:

Method 1: Automatic Rebuild (Easiest)

Cloud Run should automatically detect the new commit and start building. Wait 5-10 minutes and check if it builds successfully.

Method 2: Manual Trigger (Recommended)

To force an immediate rebuild:

1. **Go to Cloud Run Console:**
   https://console.cloud.google.com/run

2. **Click on your service:** `hotel-shift-log`

3. **Click "EDIT & DEPLOY NEW REVISION"** (button at top)

4. **Don't change anything** - just verify:
   - Repository: `jpress-knighted/Hotel_Front_Desk_Shift_Log`
   - Branch: `main` (or `master`)
   - Build type: Dockerfile

5. **Click "DEPLOY"** at the bottom

6. **Wait 5-10 minutes** for the build to complete

📊 Step 3: Monitor the Build

Watch Build Progress

Option 1: Via Cloud Run
1. In Cloud Run service page, click **“Build History”**
2. Watch the new build appear
3. Status should change from “Running” → “Success”

Option 2: Via Cloud Build
1. Go to: https://console.cloud.google.com/cloud-build/builds
2. Watch for new build to start
3. Click on it to see live logs

What Should Happen (5-10 minutes):

1. ✅ **Fetching source** - Pulls from GitHub
2. ✅ **Building** - Runs Dockerfile steps
   - Install dependencies (~2 min)
   - Generate Prisma client (~30 sec)
   - Build Next.js (~2 min)
   - Create production image (~1 min)
3. ✅ **Pushing image** - Uploads to Google Container Registry (~1 min)
4. ✅ **Deploying** - Starts new Cloud Run revision (~1 min)
5. ✅ **Success!** - New revision becomes active

✅ Step 4: Verify Deployment Succeeded

Check Build Status

1. **Cloud Build should show:** ✅ **SUCCESS** (green)
2. **Cloud Run should show:** ✅ Green checkmark on service

Check Your App URL

1. **Visit your Cloud Run URL:**
   `https://hotel-shift-log-143559442445.us-central1.run.app`

2. **You should see:**
   - ✅ Your login page (not the placeholder!)
   - ✅ Proper styling and UI
   - ✅ No console errors

3. **Test login:**
   - Username: `employee` (or `manager`)
   - Password: (the one you set)
   - Should successfully log in!

🎯 Expected Timeline

Step	Time	What's Happening
Push to GitHub	5 seconds	Uploading changes
Cloud Build triggers	30 seconds	Detects new commit
Building	5-7 minutes	Docker build process
Deploying	1-2 minutes	Rolling out new revision
Total	**~8-10 minutes**	Complete deployment

🐛 Troubleshooting

Problem: Git push fails with authentication error

Solution:

```
# If using HTTPS, you may need a personal access token
# Go to GitHub → Settings → Developer settings → Personal access tokens
# Create a token with "repo" permissions
# Use token as password when pushing
```

Or use SSH:

```
# Generate SSH key if you don't have one
ssh-keygen -t ed25519 -C "your_email@example.com"

# Add to GitHub: Settings → SSH and GPG keys → New SSH key
# Copy the public key:
cat ~/.ssh/id_ed25519.pub

# Update remote to use SSH
git remote set-url origin git@github.com:jpress-knighted/
Hotel_Front_Desk_Shift_Log.git
```

Problem: Build still fails

Check the new build logs:

1. Cloud Build → Latest build → Logs
2. Look for error messages
3. Common issues:
   - **"Cannot find module"** → Missing dependency in package.json
   - **"Prisma generate failed"** → Database schema issue (unlikely now)
   - **"Build timeout"** → Need to increase timeout (rare)

If you see new errors, share them and I'll help!

Problem: Build succeeds but app shows errors

Check Cloud Run logs:

1. Cloud Run → hotel-shift-log → LOGS tab
2. Look for runtime errors:
   - Database connection issues
   - Missing environment variables
   - Secret access problems

🎉 Success Checklist

After completing all steps, verify:

- [ ] Dockerfile and .dockerignore visible in GitHub repo root
- [ ] Cloud Build shows SUCCESS status
- [ ] Cloud Run shows green checkmark
- [ ] App URL loads login page (not placeholder)
- [ ] Can log in successfully
- [ ] No console errors in browser
- [ ] Files show proper styling

If all checked, your app is deployed! 🚀

📝 What Changed Technically

For your understanding:

Before:

```
repo/
  nextjs_space/
    Dockerfile          ← Cloud Build couldn't find this
    (all app code)
```

After:

```
repo/
  Dockerfile            ← Cloud Build finds this!
  .dockerignore
  nextjs_space/
    (all app code)
```

The new Dockerfile at the root properly references the `nextjs_space/` subdirectory, so Cloud Build can:

1. Find the Dockerfile ✅
2. Copy files from nextjs_space/ ✅
3. Build the app successfully ✅

🔄 Next: Set Up Persistent File Storage

After your app deploys successfully, don't forget to:

1. **Follow** `GCS_SETUP_GUIDE.md` to mount Google Cloud Storage
2. This ensures uploaded files persist across redeployments
3. Takes ~15-30 minutes to configure

But first, let's get your app deployed!

💬 Need Help?

If you encounter any issues:

1. **Share the error message** from Cloud Build logs
2. **Share screenshot** of the error
3. **Tell me which step failed** (push, build, or deploy)

I'll help you debug and fix it!

Current Status: ✅ Changes committed locally, ready to push!

Next Action: Push to GitHub and trigger Cloud Run rebuild!

Good luck! 🚀