# Google Cloud Platform Deployment Checklist

## ✅ Pre-Deployment Tasks

### 1. Database Cleanup

- [x] **Data cleared** - All reports, comments, and attachments removed
- [x] **Users preserved** - All user accounts remain intact
- [ ] **Clear uploads directory manually** - Remove files from `/nextjs_space/uploads/` folder
- [ ] **Change default passwords** - Update admin, manager, and employee passwords

**To clear uploads:**

```
cd /home/ubuntu/hotel_shift_log/nextjs_space/uploads
rm -rf *
```

**To change passwords (after deployment):**

```
# Generate new hash
node -e "const bcrypt = require('bcryptjs'); con-
sole.log(bcrypt.hashSync('NEW_PASSWORD', 12));"

# Update in database
# Connect to Cloud SQL and run:
UPDATE "users" SET password = '$2a$12$HASH...' WHERE username = 'admin';
```

---

## 📋 Deployment Steps

### Step 1: Create GCP Project

```
gcloud config set project YOUR_PROJECT_ID
```

### Step 2: Set Up Cloud SQL

```
# Create PostgreSQL instance
gcloud sql instances create hotel-shift-log-db \
  --database-version=POSTGRES_14 \
  --tier=db-custom-2-4096 \
  --region=us-central1 \
  --root-password=STRONG_PASSWORD \
  --storage-size=20GB \
  --storage-type=SSD \
  --backup-start-time=02:00 \
  --enable-bin-log \
  --availability-type=REGIONAL

# Create database
gcloud sql databases create hotel_shift_log \
  --instance=hotel-shift-log-db

# Create user
gcloud sql users create hoteluser \
  --instance=hotel-shift-log-db \
  --password=STRONG_PASSWORD

# Get connection name
gcloud sql instances describe hotel-shift-log-db \
  --format="value(connectionName)"
```

### Step 3: Configure Secrets

```
# Generate strong secrets
NEXTAUTH_SECRET=$(openssl rand -base64 32)
echo -n "$NEXTAUTH_SECRET" | gcloud secrets create nextauth-secret --data-file=-

# SMTP password (for Gmail, use App Password)
echo -n "YOUR_SMTP_PASSWORD" | gcloud secrets create smtp-password --data-file=-

# Grant access
PROJECT_NUMBER=$(gcloud projects describe YOUR_PROJECT_ID --format="value(projectNumbe
r)")
gcloud secrets add-iam-policy-binding nextauth-secret \
  --member="serviceAccount:${PROJECT_NUMBER}-compute@developer.gserviceaccount.com" \
  --role="roles/secretmanager.secretAccessor"

gcloud secrets add-iam-policy-binding smtp-password \
  --member="serviceAccount:${PROJECT_NUMBER}-compute@developer.gserviceaccount.com" \
  --role="roles/secretmanager.secretAccessor"
```

## Step 4: Build and Deploy

```
# Enable APIs
gcloud services enable cloudbuild.googleapis.com run.googleapis.com artifactre-
gistry.googleapis.com

# Create registry
gcloud artifacts repositories create hotel-shift-log \
  --repository-format=docker \
  --location=us-central1

# Build image
cd /home/ubuntu/hotel_shift_log/nextjs_space
gcloud builds submit \
  --tag us-central1-docker.pkg.dev/YOUR_PROJECT_ID/hotel-shift-log/app:v1.0.0

# Deploy to Cloud Run
gcloud run deploy hotel-shift-log \
  --image us-central1-docker.pkg.dev/YOUR_PROJECT_ID/hotel-shift-log/app:v1.0.0 \
  --platform managed \
  --region us-central1 \
  --allow-unauthenticated \
  --memory 2Gi \
  --cpu 2 \
  --min-instances 1 \
  --max-instances 10 \
  --add-cloudsql-instances YOUR_CONNECTION_NAME \
  --set-env-vars "DATABASE_URL=postgresql://hoteluser:PASSWORD@/hotel_shift_log?host=/
cloudsql/YOUR_CONNECTION_NAME" \
  --set-secrets "NEXTAUTH_SECRET=nextauth-secret:latest,SMTP_PASSWORD=smtp-pass-
word:latest" \
  --set-env-vars "NEXTAUTH_URL=https://
YOUR_DOMAIN.com,SMTP_HOST=smtp.gmail.com,SMTP_PORT=587,SMTP_USER=your-email@gmail.com"
```

## Step 5: Run Database Migrations

```
# Start Cloud SQL proxy
cloud-sql-proxy YOUR_CONNECTION_NAME &

# Set database URL
export DATABASE_URL="postgresql://hoteluser:PASSWORD@localhost:5432/hotel_shift_log"

# Run migrations
cd /home/ubuntu/hotel_shift_log/nextjs_space
npx prisma db push

# Seed database
npx prisma db seed

# Stop proxy
kill %1
```

## Step 6: Configure Email Recipients

1. Log into the deployed application as admin
2. Go to **Users** page
3. For each manager who should receive high-priority alerts:
   - Click **Edit**

- Toggle **Receives High Priority Emails** to ON
- Ensure email address is filled in
- Click **Save**

### Step 7: Set Up Custom Domain (Optional)

```
# Map domain
gcloud run domain-mappings create \
  --service hotel-shift-log \
  --domain your-domain.com \
  --region us-central1

# Follow DNS verification instructions
```

### Step 8: Configure Backups

```
# Create backup bucket
gcloud storage buckets create gs://YOUR_PROJECT_ID-hotel-backups \
  --location=us-central1 \
  --uniform-bucket-level-access

# Enable versioning
gcloud storage buckets update gs://YOUR_PROJECT_ID-hotel-backups \
  --versioning

# Set retention (90 days)
cat > lifecycle.json << EOF
{
  "lifecycle": {
    "rule": [{
      "action": {"type": "Delete"},
      "condition": {"age": 90}
    }]
  }
}
EOF

gcloud storage buckets update gs://YOUR_PROJECT_ID-hotel-backups \
  --lifecycle-file=lifecycle.json
```

---

## 🔒 Security Hardening

### Change Default Passwords

- [ ] Admin password changed
- [ ] Manager password changed
- [ ] Employee password changed
- [ ] Database password is strong (20+ characters)
- [ ] NEXTAUTH_SECRET is strong (32+ characters)

### Configure Monitoring

```
# Set up uptime checks
gcloud monitoring uptime-checks create http hotel-shift-log \
  --resource-type=url \
  --url="https://YOUR_DOMAIN.com/login"

# Set up error rate alerts
# (Create through Cloud Console UI)
```

### Enable Cloud Armor (Optional)

```
# Create WAF policy
gcloud compute security-policies create hotel-waf \
  --description "WAF for hotel shift log"

# Add rate limiting
gcloud compute security-policies rules create 1000 \
  --security-policy hotel-waf \
  --expression "true" \
  --action "rate-based-ban" \
  --rate-limit-threshold-count 100 \
  --rate-limit-threshold-interval-sec 60
```

---

## 🧪 Post-Deployment Testing

### Functionality Tests

- [ ] Can log in with all three roles
- [ ] Employees can create reports
- [ ] Managers can add comments
- [ ] File uploads work correctly
- [ ] High-priority reports send emails
- [ ] PDF/CSV export works
- [ ] User management works (admin only)
- [ ] Archive/unarchive functionality works
- [ ] Report acknowledgement works
- [ ] Comment likes work

### Security Tests

- [ ] Cannot access dashboard without login
- [ ] Archived users cannot log in
- [ ] Managers cannot create super admins
- [ ] Employees cannot see manager notes
- [ ] File size limits are enforced
- [ ] Daily post limits are enforced
- [ ] Path traversal blocked (try `../../etc/passwd` in file serving)
- [ ] XSS blocked (try `<script>alert('xss')</script>` in text fields)

### Performance Tests

- [ ] Page load time < 3 seconds
- [ ] Large file uploads work (up to 30MB)
- [ ] Can handle 10 concurrent users
- [ ] Database queries are fast

---

## 📧 Email Configuration

### Gmail Setup (Testing)

1. Enable 2FA on Gmail account
2. Generate App Password: https://myaccount.google.com/apppasswords
3. Use in SMTP_PASSWORD

### SendGrid Setup (Production)

1. Create account: https://sendgrid.com/
2. Verify domain
3. Generate API key
4. Configure:
   `SMTP_HOST=smtp.sendgrid.net`
   `SMTP_PORT=587`
   `SMTP_USER=apikey`
   `SMTP_PASSWORD=SG.xxxxxxxxxxx`

### Test Email

```
# Create a high-priority report and verify email is received
```

---

## 💾 Backup Procedures

### Manual Backup

```
# Database
gcloud sql backups create --instance=hotel-shift-log-db

# Files
gcloud storage cp -r /path/to/uploads gs://YOUR_PROJECT_ID-hotel-backups/manual-$
(date +%Y%m%d)/
```

### Restore Procedure

```
# List backups
gcloud sql backups list --instance=hotel-shift-log-db

# Restore database
gcloud sql backups restore BACKUP_ID \
  --backup-instance=hotel-shift-log-db \
  --target-instance=hotel-shift-log-db

# Restore files
gcloud storage cp -r gs://YOUR_PROJECT_ID-hotel-backups/backup-YYYYMMDD/* /path/to/up-
loads/
```

---

## 📊 Monitoring Checklist

### Set Up Alerts For:

- [ ] Error rate > 5%
- [ ] Response time > 5 seconds
- [ ] CPU usage > 80%
- [ ] Memory usage > 80%
- [ ] Disk usage > 80%
- [ ] Failed login attempts > 10 per hour
- [ ] Database connection failures

### Regular Reviews:

- [ ] **Daily**: Check error logs
- [ ] **Weekly**: Review backup success
- [ ] **Monthly**: Review user access, test backups
- [ ] **Quarterly**: Security audit, performance review

---

## 🚨 Incident Response

### If Unauthorized Access Detected:

1. Immediately revoke all active sessions
2. Change all passwords
3. Review audit logs
4. Enable Cloud SQL read-only mode temporarily
5. Investigate and patch vulnerability
6. Notify affected parties

### If Data Loss Occurs:

1. Stop all write operations
2. Identify last known good backup
3. Restore from backup to separate instance

4. Verify data integrity
5. Switch to restored instance
6. Investigate root cause

---

## 📝 Documentation Links

- **Full README**: `/home/ubuntu/hotel_shift_log/README.md`
- **Security Analysis**: `/home/ubuntu/hotel_shift_log/SECURITY.md`
- **GCP Console**: https://console.cloud.google.com
- **Cloud SQL**: https://console.cloud.google.com/sql
- **Cloud Run**: https://console.cloud.google.com/run
- **Secret Manager**: https://console.cloud.google.com/security/secret-manager

---

## ✅ Sign-Off

### Before Going Live:

- [ ] All pre-deployment tasks completed
- [ ] All deployment steps completed
- [ ] All post-deployment tests passed
- [ ] Email notifications tested
- [ ] Backups configured and tested
- [ ] Monitoring and alerts configured
- [ ] Security hardening completed
- [ ] Documentation reviewed
- [ ] Team trained on system usage
- [ ] Incident response plan in place

**Deployed By**: __
**Date**: __
**Deployment URL**: __
**Database Instance**: __
**Backup Location**: ___

---

## 📞 Support Contacts

**System Administrator**: __
**GCP Support**: https://cloud.google.com/support
**Emergency Contact**: __

---

**Document Version**: 1.0
**Last Updated**: October 21, 2025