Cyber Trust Sensor Dashboard

VPS Deployment Reference Guide

Quick Reference Card

VPS Details:

- **IP Address:** `31.97.114.80`
- **Dashboard Location:** `/root/my-working-prototype-dashbaord/`
- **User:** `root`
- **Main Port:** `80` (HTTP), `443` (HTTPS)

📋 DAILY OPERATIONS CHECKLIST

Starting Your Day

```bash
ssh root@31.97.114.80
cd /root/my-working-prototype-dashbaord
docker ps  # Check what's running
```

Before Any Update

1. ✅ Check services are running: `docker ps`
2. ✅ Create backup: `tar -czf backup-$(date +%Y%m%d).tar.gz build/`
3. ✅ Note current version: `git log -1 --oneline` (if using git)

🚀 DEPLOYMENT WORKFLOWS

Workflow 1: Frontend Update (Most Common)

On Your Local Machine:

```bash
```

```bash
# 1. Make your changes
# 2. Test locally
npm start

# 3. Build production version
npm run build

# 4. Deploy to VPS
scp -r ./build/* root@31.97.114.80:/root/my-working-prototype-dashbaord/build/
```

Result: Changes are live immediately! No restart needed for static files.

Workflow 2: Full Stack Update

On Your Local Machine:

```bash
# 1. Build everything
npm run build

# 2. Transfer all files
scp -r ./* root@31.97.114.80:/root/my-working-prototype-dashbaord/
```

On VPS:

```bash
ssh root@31.97.114.80
cd /root/my-working-prototype-dashbaord
docker compose down
docker compose up -d
```

Workflow 3: Quick Fix

For urgent fixes without local build:

```bash
ssh root@31.97.114.80
cd /root/my-working-prototype-dashbaord
nano src/components/ProblemComponent.js  # Edit directly
npm run build  # Build on VPS
docker compose restart nginx
```

📁 DIRECTORY STRUCTURE

```
/root/my-working-prototype-dashbaord/
├── build/              # 🎯 Production files (deploy here!)
├── src/                # Source code
├── public/             # Public assets
├── node_modules/       # Dependencies (don't touch)
├── nginx/              # Nginx configuration
│   └── conf.d/
│       └── default.conf  # Main config
├── docker-compose.yml   # Service definitions
├── package.json        # Project config
└── .env                # Environment variables
```

🐳 DOCKER COMMANDS

Essential Commands

Command	Purpose
`docker ps`	Show running containers
`docker ps -a`	Show all containers
`docker compose up -d`	Start all services
`docker compose down`	Stop all services
`docker compose restart`	Restart all services
`docker compose logs`	View logs
`docker compose logs -f nginx`	Follow nginx logs

Container Names

- **nginx** - Web server (serves your app)
- **api** - Backend API (if configured)
- **postgres/db** - Database (if configured)

🔧 TROUBLESHOOTING GUIDE

Problem: Site Not Loading

Check 1: Are containers running?

```bash
```

```
docker ps
```

❌ If no containers → `docker compose up -d`

Check 2: Is port 80 accessible?

```bash
netstat -tuln | grep :80
```

❌ If not listening → Check nginx: `docker logs nginx`

Check 3: Does build exist?

```bash
ls -la /root/my-working-prototype-dashbaord/build/
```

❌ If empty → Deploy build files from local

Problem: Changes Not Showing

Solution 1: Clear browser cache

- Hard refresh: `Ctrl+Shift+R` (Windows) or `Cmd+Shift+R` (Mac)

Solution 2: Restart nginx

```bash
docker compose restart nginx
```

Solution 3: Check file timestamps

```bash
ls -la build/ | head -5
```

Old dates = files weren't copied

Problem: Container Won't Start

Check logs:

```bash
```

```
docker compose logs [container_name]
```

Common fixes:

```bash
# Port conflict
sudo lsof -i :80  # See what's using port 80
kill -9 [PID]     # Kill conflicting process

# Permission issue
chmod -R 755 /root/my-working-prototype-dashbaord/

# Corrupted container
docker compose down
docker system prune -f
docker compose up -d
```

📊 MONITORING & LOGS

View Real-Time Logs

```bash
# All containers
docker compose logs -f

# Specific container
docker logs -f nginx

# Last 100 lines
docker logs --tail 100 nginx
```

Check Resource Usage

```bash
```

```
# Container stats
docker stats

# Disk space
df -h

# Memory
free -h

# CPU
top
```

Log Files Location

- Docker logs: `/var/lib/docker/containers/*/`
- App logs: `/root/my-working-prototype-dashbaord/logs/`
- System logs: `/var/log/`

🔐 SECURITY CHECKLIST

Weekly Tasks

- [ ] Check for unusual login attempts: `last -20`
- [ ] Review docker logs for errors
- [ ] Update system: `apt update && apt upgrade`
- [ ] Check disk space: `df -h`

Monthly Tasks

- [ ] Backup database (if applicable)
- [ ] Review and rotate logs
- [ ] Check SSL certificate expiry
- [ ] Update dependencies: `npm audit`

💾 BACKUP & RESTORE

Create Backup

```
bash
```

```
cd /root/my-working-prototype-dashbaord
tar -czf ~/backup-$(date +%Y%m%d).tar.gz \
  --exclude=node_modules \
  --exclude=.git \
  .
```

Restore Backup

```bash
cd /root/my-working-prototype-dashbaord
tar -xzf ~/backup-20250808.tar.gz
docker compose restart
```

Automated Backup (Add to crontab)

```bash
# Edit crontab
crontab -e

# Add daily backup at 2 AM
0 2 * * * cd /root/my-working-prototype-dashbaord && tar -czf ~/backups/backup-$(date +\%Y\%m\%d).tar.gz --excl
```

🚦 QUICK STATUS CHECKS

Is Everything Working?

```bash
curl http://localhost/health
# Should return: OK or {"status":"UP"}
```

Check From Outside

```bash
curl http://31.97.114.80
# Should return your HTML
```

Test API (if configured)

```bash
```

```
curl http://localhost/api/status
# Should return JSON status
```

📝 COMMON TASKS SCRIPTS

One-Line Deploy

```bash
# Add to ~/.bashrc for quick access
alias deploy='scp -r ./build/* root@31.97.114.80:/root/my-working-prototype-dashbaord/build/'
```

Quick Connect

```bash
# Add to local ~/.ssh/config
Host dashboard
    HostName 31.97.114.80
    User root
    Port 22

# Then just: ssh dashboard
```

Emergency Reset

```bash
#!/bin/bash
# Save as reset.sh on VPS
cd /root/my-working-prototype-dashbaord
docker compose down
docker system prune -f
docker compose up -d
echo "Reset complete!"
```

📞 SUPPORT INFORMATION

System Details

- **OS:** Ubuntu 24.04.2 LTS
- **Docker:** Version 28.3.3
- **Docker Compose:** v2.39.1

- **Node.js:** Check with `node --version`
- **NPM:** Check with `npm --version`

Key File Locations

- **Main App:** `/root/my-working-prototype-dashbaord/`
- **Docker Config:** `/root/my-working-prototype-dashbaord/docker-compose.yml`
- **Nginx Config:** `/root/my-working-prototype-dashbaord/nginx/conf.d/default.conf`
- **Environment:** `/root/my-working-prototype-dashbaord/.env`
- **Logs:** `/var/log/dashboard-deployment.log`

When to Restart Services

No Restart Needed:

- ✅ Updating files in `build/` directory
- ✅ Changing static assets (images, CSS, JS)
- ✅ Updating HTML files

Restart Required:

- ⚠️ Changing `docker-compose.yml`
- ⚠️ Updating `.env` variables
- ⚠️ Modifying nginx configuration
- ⚠️ Updating API code
- ⚠️ Database schema changes

📈 PERFORMANCE OPTIMIZATION

Quick Wins

1. **Enable Gzip in nginx:**

```nginx
gzip on;
gzip_types text/plain text/css application/json application/javascript;
```

2. **Set Cache Headers:**

```nginx
```

```
location ~* \.(jpg|jpeg|png|gif|ico|css|js)$ {
    expires 1y;
    add_header Cache-Control "public, immutable";
}
```

3. **Optimize Images:**

```bash
# Install image optimizer
apt install jpegoptim optipng
# Optimize
find build -name "*.jpg" -exec jpegoptim {} \;
find build -name "*.png" -exec optipng {} \;
```

🎯 DEPLOYMENT DECISION TREE

```
Need to update something?
│
├─ Is it just HTML/CSS/JS in build/?
│   └─ YES → Copy files → Done! ✅
│   └─ NO → Continue ↓
│
├─ Is it React component code?
│   └─ YES → Build locally → Copy build/ → Done! ✅
│   └─ NO → Continue ↓
│
├─ Is it API/Backend code?
│   └─ YES → Copy files → docker compose restart api ✅
│   └─ NO → Continue ↓
│
├─ Is it configuration (nginx, docker)?
│   └─ YES → Copy files → docker compose down → up -d ✅
│   └─ NO → Continue ↓
│
└─ Database change?
   └─ YES → Backup first! → Run migration → Restart API ✅
```

📚 ADDITIONAL RESOURCES

Documentation

- Docker: https://docs.docker.com

- Docker Compose: https://docs.docker.com/compose
- Nginx: https://nginx.org/en/docs
- React: https://react.dev

Monitoring Tools

- Server Monitoring: `htop` (install with `apt install htop`)
- Network Monitoring: `nethogs` (install with `apt install nethogs`)
- Docker Monitoring: `ctop` (https://github.com/bcicen/ctop)

Document Version: 2.0
Last Updated: August 2025
For: Cyber Trust Sensor Dashboard
Location: `/root/my-working-prototype-dashbaord/`
IP: `31.97.114.80`

Keep this guide handy for quick reference during deployments!