
Summary

This VPS hosts **Kerit**, a Node.js/TypeScript-based web application. It runs under PM2 (cluster mode), serves traffic via Nginx (with Let's Encrypt SSL), uses PostgreSQL as its database, and features an automatic deployment webhook. This documentation explains system details, directory structure, service configuration, deployment processes, and maintenance procedures.

Basic System Info

Item	Value
Hostname	kerit-vps
OS	Ubuntu 24.04.1 LTS (Noble)
Kernel	Linux 6.8.0 (x86_64)
Primary User	kerit
Public Domains	kerit.com.ru, www.kerit.com.ru
Web Ports	80 (HTTP), 443 (HTTPS)
Firewall	UFW inactive (consider enabling)

Application Stack

Component	Description
Node.js	JavaScript runtime (v20.x from NodeSource)
PM2	Process manager (`cluster` mode, 2 instances)
PostgreSQL	Database server (with `kerit_db` database and `kerit_user` role)
Nginx	Reverse proxy & TLS termination
Certbot	Let's Encrypt SSL certificate management
Git	Source control
Express.js	Web framework (used for webhook server)

Directory and File Structure

```
/home/kerit/
├── deploy.sh                  # Deployment script
├── health-check.sh            # Health check script
├── logs/
│   ├── kerit-error.log        # Kerit app stderr
│   ├── kerit-out.log          # Kerit app stdout
│   ├── kerit-combined.log     # Combined PM2 logs
│   ├── webhook-error.log      # Webhook server stderr
│   ├── webhook-out.log        # Webhook server stdout
│   └── deploy.log             # Deployment run logs
├── projects/
│   └── kerit-app/
│       ├── .env               # Environment variables (600)
│       ├── ecosystem.config.json # PM2 process file
│       ├── dist/              # Compiled output
│       ├── node_modules/      # Dependencies
│       ├── src/               # Source code
│       └── ...
└── webhook-server.js          # Express webhook listener
└── webhook.ecosystem.config.js # PM2 config for webhook
```

How the App Works

- **PM2 Startup**: Managed via `ecosystem.config.json` (app name `kerit-app`, 2 instances, cluster mode).
- **Node Server**: Listens on `localhost:3001` serving the built Express/Next.js app.
- **Database**: PostgreSQL database `kerit_db`, user `kerit_user` with password managed in `.env`.
- **Deployment Webhook**: Express server on port `3002` verifies GitHub signatures and runs `deploy.sh` as the `kerit` user.

PM2 Commands

```
pm2 start ecosystem.config.json   # Start/Restart app
pm2 start webhook.ecosystem.config.js  # Start webhook server
pm2 save
pm2 startup  # Generate startup script for systemd
```

Environment Configuration

/home/kerit/projects/kerit-app/.env (permissions `600`)

```
NODE_ENV=production
PORT=3001
DATABASE_URL=postgresql://kerit_user:ferfgrtgSGSERGERG423423FEFERFERF@localhost:
5432/kerit_db
SESSION_SECRET=nrsjvlsjdvfgTREHBRSGTWRGTefhewurbcfewrWCERCGWERCGwegethytjbuikUIKYUINKUYxfwefwexf
REPL_ID=kerit-vps-production
REPLIT_DOMAINS=kerit.com.ru
ISSUER_URL=https://replit.com/oidc
```

Nginx Configuration

/etc/nginx/sites-available/kerit

```
# Redirect HTTP to HTTPS
server {
    listen 80;
    server_name kerit.com.ru www.kerit.com.ru;
    return 301 https://$host$request_uri;
}

# HTTPS server
server {
    listen 443 ssl http2;
    server_name kerit.com.ru www.kerit.com.ru;

    ssl_certificate /etc/letsencrypt/live/kerit.com.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/kerit.com.ru/privkey.pem;

    add_header X-Frame-Options DENY;
    add_header X-Content-Type-Options nosniff;
    add_header X-XSS-Protection "1; mode=block";
    add_header Strict-Transport-Security "max-age=63072000; includeSubDomains;
preload";

    gzip on;
    gzip_vary on;
    gzip_min_length 1024;
    gzip_types text/plain text/css text/xml text/javascript application/
javascript application/xml+rss application/json;
```

```
    location / {
        proxy_pass http://127.0.0.1:3001;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
        proxy_read_timeout 300s;
        proxy_connect_timeout 75s;
    }

    location /webhook {
        proxy_pass http://127.0.0.1:3002;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location ~* \.(js|css|png|jpg|jpeg|gif|ico|svg|woff|woff2|ttf|eot)$ {
        proxy_pass http://127.0.0.1:3001;
        expires 1y;
        add_header Cache-Control "public, immutable";
    }
}
```

Enable with:

```
sudo ln -s /etc/nginx/sites-available/kerit /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```

SSL Certificate

Managed by Certbot:

```
sudo certbot --nginx -d kerit.com.ru -d www.kerit.com.ru
sudo systemctl reload nginx
```

Certificates auto-renew by default (via cron/systemd).

Deployment Script

/home/kerit/deploy.sh

```
#!/bin/bash
set -e
PROJECT_DIR="/home/kerit/projects/kerit-app"
LOG_FILE="/home/kerit/logs/deploy.log"

log() { echo "$(date): $1" | tee -a $LOG_FILE; }

log "Starting deployment..."
cd $PROJECT_DIR

log "Creating backup..."
cp -r $PROJECT_DIR $PROJECT_DIR.backup.$(date +%Y%m%d_%H%M%S)

log "Pulling latest changes from GitHub..."
git pull origin main

log "Installing dependencies..."
npm ci --production

log "Building application..."
npm run build

log "Updating database schema..."
npm run db:push

log "Restarting application..."
pm run pm2 reload kerit-app

sleep 10
if pm2 list | grep -q "kerit-app.*online"; then
    log "Deployment successful!"
    ls -t $PROJECT_DIR.backup.* | tail -n +6 | xargs -r rm -rf
else
    log "Deployment failed! Rolling back..."
    pm2 stop kerit-app
    rm -rf $PROJECT_DIR
    mv $(ls -t $PROJECT_DIR.backup.* | head -1) $PROJECT_DIR
    cd $PROJECT_DIR
    pm2 start kerit-app
```

```
    exit 1
fi

log "Deployment completed successfully!"
```

Make executable and owned by `kerit`:

```
chmod +x /home/kerit/deploy.sh
chown kerit:kerit /home/kerit/deploy.sh
```

Webhook Server

/home/kerit/webhook-server.js

```
const express = require('express');
const crypto = require('crypto');
const { exec } = require('child_process');

const app = express();
const PORT = 3002;
const SECRET = 'csdcdf23423$@#$FEFREHY';

app.use(express.json());

function verifySignature(req) {
    const signature = req.headers['x-hub-signature-256'];
    const hmac = crypto.createHmac('sha256', SECRET);
    const digest = 'sha256=' +
hmac.update(JSON.stringify(req.body)).digest('hex');
    return crypto.timingSafeEqual(Buffer.from(digest), Buffer.from(signature));
}

app.post('/webhook', (req, res) => {
    if (!verifySignature(req)) return res.status(401).send('Unauthorized');
    if (req.body.ref === 'refs/heads/main') {
        exec('sudo -u kerit /home/kerit/deploy.sh', (err, stdout, stderr) => {
            if (err) return res.status(500).send('Deployment failed');
            console.log(stdout);
            res.send('Deployment triggered');
        });
    } else res.send('No deployment needed');
});
```

```
app.listen(PORT, () => console.log(`Webhook server running on port ${PORT}`));
```

Managed via PM2 with `webhook.ecosystem.config.js`.

Health Checks & Monitoring

- **Manual check:** `/home/kerit/health-check.sh`
- **Cron job:** *(if configured)* runs health-check script periodically to restart any downed PM2 apps.
- **PM2 status:** `pm2 status`
- **System logs:**
- Nginx: `/var/log/nginx/access.log`, `/var/log/nginx/error.log`
- Deployment: `/home/kerit/logs/deploy.log`

Security Notes

- **Firewall**: UFW installed but inactive. Enable with:

```
sudo ufw allow OpenSSH
sudo ufw allow 'Nginx Full'
sudo ufw enable
```

- **SSH**: Use key-based auth; disable password login.
- **SSL**: Managed by Certbot; auto-renewal enabled.
- **Process User**: All services run under non-root user `kerit`.

Current Weak Points

- UFW is inactive.
- No automated cron configured for health-check script.
- Secrets stored in plaintext `.env`; consider a vault/encrypted store.

Recommended Improvements

- Enable and configure UFW.
- Set up a cron or systemd timer for `health-check.sh` (e.g., every 5 minutes).
- Integrate a secret management solution (HashiCorp Vault or AWS Secrets Manager).
- Add monitoring/alerting (Prometheus + Grafana, or external service).

For New Developers or Admins

Starting the App:

```
sudo pm2 start /home/kerit/projects/kerit-app/ecosystem.config.json
sudo pm2 save
```

Checking Logs:

```
tail -f /home/kerit/logs/kerit-combined.log
```

Reloading Nginx:

```
sudo nginx -t && sudo systemctl reload nginx
```

Renewing SSL:

```
sudo certbot renew --dry-run
```

Summary

The **Kerit** application is deployed with a modern Node.js stack, managed by PM2, proxied by Nginx with HTTPS, and backed by PostgreSQL. A webhook-driven deployment pipeline with rollback safeguards ensures continuous delivery. Security hardening and monitoring recommendations are provided to maintain reliability and resilience.