**🚀 Complete Deployment Guide for IbizaGirl.pics**

**📋 Table of Contents**

1. [Pre-Deployment Checklist](#pre-deployment-checklist)
2. [Server Requirements](#server-requirements)
3. [Step-by-Step Deployment](#step-by-step-deployment)
4. [Security Configuration](#security-configuration)
5. [PayPal Setup](#paypal-setup)
6. [CDN Configuration](#cdn-configuration)
7. [SSL/HTTPS Setup](#ssl-https-setup)
8. [Monitoring & Maintenance](#monitoring-maintenance)
9. [Troubleshooting](#troubleshooting)

**📋 Pre-Deployment Checklist**

**Required Files Structure**

ibizagirl.pics/

├── public/

│ ├── index.html (main file)

│ ├── assets/

│ │ ├── censored/ # 213 blurred images

│ │ ├── censored-videos/ # 10 blurred videos

│ │ ├── full/ # 36 teaser images (visible)

│ │ ├── uncensored/ # Full resolution images (protected)

│ │ └── videos/ # Full videos (protected)

│ ├── style.css

│ ├── script.js

│ ├── paypal-payments.js

│ └── gallery.json

├── server/

│ ├── app.js

│ ├── package.json

│ └── .env

├── backup/

└── logs/

**Environment Variables (.env)**

# Production Environment

NODE\_ENV=production

PORT=3000

# Security Keys (Generate new ones!)

JWT\_SECRET=generate-64-character-random-string-here

ENCRYPTION\_KEY=generate-32-byte-hex-key-here

# PayPal Live Credentials

PAYPAL\_CLIENT\_ID=AfQEdiielw5fm3wF08p9pcxwqR3gPz82YRNUTKY4A8WNG9AktiGsDNyr2i7BsjVzSwwpeCwR7Tt7DPq5

PAYPAL\_SECRET=your-paypal-live-secret-here

PAYPAL\_WEBHOOK\_ID=your-webhook-id-here

PAYPAL\_SUBSCRIPTION\_PLAN\_ID=GCUY49PRUB6V2

# Database

DATABASE\_URL=mongodb://username:password@localhost:27017/ibizagirl

REDIS\_URL=redis://localhost:6379

# CDN & Domain

CDN\_URL=https://cdn.ibizagirl.pics

DOMAIN=https://ibizagirl.pics

# Email (for notifications)

SMTP\_HOST=smtp.gmail.com

SMTP\_PORT=587

SMTP\_USER=your-email@gmail.com

SMTP\_PASS=your-app-password

**🖥️ Server Requirements**

**Minimum Specifications**

* **VPS/Dedicated Server**: Ubuntu 20.04 LTS or newer
* **RAM**: 2GB minimum (4GB recommended)
* **Storage**: 50GB SSD minimum
* **CPU**: 2 cores minimum
* **Bandwidth**: Unlimited preferred
* **SSL Certificate**: Required

**Software Requirements**

# Required packages

- Node.js 18.x or newer

- MongoDB 5.0 or newer

- Redis 6.x or newer

- Nginx 1.18 or newer

- PM2 (process manager)

- Git

- UFW (firewall)

- Certbot (Let's Encrypt SSL)

**📦 Step-by-Step Deployment**

**Step 1: Server Setup (Ubuntu 20.04 LTS)**

# Update system

sudo apt update && sudo apt upgrade -y

# Install essential packages

sudo apt install -y curl wget git build-essential ufw

# Install Node.js 18.x

curl -fsSL https://deb.nodesource.com/setup\_18.x | sudo -E bash -

sudo apt install -y nodejs

# Install MongoDB

wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list

sudo apt update

sudo apt install -y mongodb-org

sudo systemctl start mongod

sudo systemctl enable mongod

# Install Redis

sudo apt install -y redis-server

sudo systemctl enable redis-server

# Install Nginx

sudo apt install -y nginx

# Install PM2 globally

sudo npm install -g pm2

# Install Certbot for SSL

sudo apt install -y certbot python3-certbot-nginx

**Step 2: Configure Firewall**

# Setup UFW firewall

sudo ufw default deny incoming

sudo ufw default allow outgoing

sudo ufw allow ssh

sudo ufw allow 80

sudo ufw allow 443

sudo ufw enable

**Step 3: Create Application User**

# Create dedicated user for security

sudo adduser ibizaapp --disabled-password

sudo usermod -aG sudo ibizaapp

# Switch to app user

sudo su - ibizaapp

**Step 4: Deploy Application**

# Clone your repository (or upload files)

cd /home/ibizaapp

git clone https://github.com/yourusername/ibizagirl.git

cd ibizagirl

# Create necessary directories

mkdir -p backup logs secure-content

# Move sensitive content to secure location

mv public/assets/uncensored secure-content/

mv public/assets/videos secure-content/

# Install dependencies

cd server

npm install --production

# Create .env file

nano .env

# (Paste your environment variables here)

# Set correct permissions

chmod 600 .env

chmod -R 755 ../public

chmod -R 700 ../secure-content

**Step 5: Configure Nginx**

Create Nginx configuration:

sudo nano /etc/nginx/sites-available/ibizagirl.pics

Paste this configuration:

# Rate limiting

limit\_req\_zone $binary\_remote\_addr zone=general:10m rate=10r/s;

limit\_req\_zone $binary\_remote\_addr zone=api:10m rate=5r/s;

limit\_req\_zone $binary\_remote\_addr zone=payment:10m rate=1r/s;

# Upstream Node.js app

upstream ibizaapp {

server 127.0.0.1:3000;

keepalive 64;

}

server {

listen 80;

server\_name ibizagirl.pics www.ibizagirl.pics;

return 301 https://$server\_name$request\_uri;

}

server {

listen 443 ssl http2;

server\_name ibizagirl.pics www.ibizagirl.pics;

# SSL Configuration

ssl\_certificate /etc/letsencrypt/live/ibizagirl.pics/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/ibizagirl.pics/privkey.pem;

ssl\_protocols TLSv1.2 TLSv1.3;

ssl\_ciphers HIGH:!aNULL:!MD5;

ssl\_prefer\_server\_ciphers on;

ssl\_session\_cache shared:SSL:10m;

ssl\_session\_timeout 10m;

# Security Headers

add\_header X-Frame-Options "SAMEORIGIN" always;

add\_header X-Content-Type-Options "nosniff" always;

add\_header X-XSS-Protection "1; mode=block" always;

add\_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;

add\_header Content-Security-Policy "default-src 'self' https://www.paypal.com https://www.paypalobjects.com; script-src 'self' 'unsafe-inline' https://www.paypal.com https://www.paypalobjects.com; style-src 'self' 'unsafe-inline';" always;

# Root directory

root /home/ibizaapp/ibizagirl/public;

index index.html;

# Gzip compression

gzip on;

gzip\_vary on;

gzip\_min\_length 1024;

gzip\_types text/plain text/css text/xml text/javascript application/json application/javascript application/xml+rss;

# Cache static files

location ~\* \.(jpg|jpeg|png|gif|ico|css|js|svg|woff|woff2|ttf|eot)$ {

expires 30d;

add\_header Cache-Control "public, immutable";

access\_log off;

}

# Protect sensitive directories

location ~ ^/(secure-content|backup|logs|server)/ {

deny all;

return 404;

}

# API endpoints with rate limiting

location /api/ {

limit\_req zone=api burst=20 nodelay;

proxy\_pass http://ibizaapp;

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;

}

# Payment endpoints with strict rate limiting

location /api/payment/ {

limit\_req zone=payment burst=2 nodelay;

proxy\_pass http://ibizaapp;

proxy\_http\_version 1.1;

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;

}

# PayPal webhooks

location /api/webhooks/paypal {

proxy\_pass http://ibizaapp;

proxy\_http\_version 1.1;

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\_set\_header Content-Type $content\_type;

proxy\_set\_header X-PayPal-Transmission-Id $http\_paypal\_transmission\_id;

proxy\_set\_header X-PayPal-Transmission-Time $http\_paypal\_transmission\_time;

proxy\_set\_header X-PayPal-Transmission-Sig $http\_paypal\_transmission\_sig;

proxy\_set\_header X-PayPal-Cert-Url $http\_paypal\_cert\_url;

proxy\_set\_header X-PayPal-Auth-Algo $http\_paypal\_auth\_algo;

}

# Block access to hidden files

location ~ /\. {

deny all;

access\_log off;

log\_not\_found off;

}

# Custom error pages

error\_page 404 /404.html;

error\_page 500 502 503 504 /50x.html;

# Logging

access\_log /var/log/nginx/ibizagirl\_access.log;

error\_log /var/log/nginx/ibizagirl\_error.log;

}

Enable the site:

sudo ln -s /etc/nginx/sites-available/ibizagirl.pics /etc/nginx/sites-enabled/

sudo nginx -t

sudo systemctl reload nginx

**Step 6: SSL Certificate Setup**

# Obtain SSL certificate

sudo certbot --nginx -d ibizagirl.pics -d www.ibizagirl.pics

# Auto-renewal

sudo systemctl enable certbot.timer

**Step 7: Start Application with PM2**

cd /home/ibizaapp/ibizagirl/server

# Create PM2 ecosystem file

nano ecosystem.config.js

module.exports = {

apps: [{

name: 'ibizagirl',

script: './app.js',

instances: 2,

exec\_mode: 'cluster',

env: {

NODE\_ENV: 'production',

PORT: 3000

},

error\_file: '../logs/error.log',

out\_file: '../logs/out.log',

log\_file: '../logs/combined.log',

time: true,

max\_memory\_restart: '1G',

cron\_restart: '0 0 \* \* \*'

}]

};

# Start application

pm2 start ecosystem.config.js

# Save PM2 configuration

pm2 save

# Setup PM2 to start on boot

pm2 startup

# (Follow the command it outputs)

**🔒 Security Configuration**

**Step 1: Secure MongoDB**

# Connect to MongoDB

mongosh

# Create admin user

use admin

db.createUser({

user: "admin",

pwd: "STRONG\_PASSWORD\_HERE",

roles: ["root"]

})

# Create application database and user

use ibizagirl

db.createUser({

user: "ibizaapp",

pwd: "ANOTHER\_STRONG\_PASSWORD",

roles: ["readWrite"]

})

# Enable authentication

exit

# Edit MongoDB config

sudo nano /etc/mongod.conf

# Add:

# security:

# authorization: enabled

sudo systemctl restart mongod

**Step 2: Secure Redis**

# Edit Redis config

sudo nano /etc/redis/redis.conf

# Set password

requirepass YOUR\_REDIS\_PASSWORD

# Bind to localhost only

bind 127.0.0.1 ::1

# Disable dangerous commands

rename-command FLUSHDB ""

rename-command FLUSHALL ""

rename-command CONFIG ""

sudo systemctl restart redis

**Step 3: File Permissions**

# Set correct ownership

sudo chown -R ibizaapp:ibizaapp /home/ibizaapp/ibizagirl

# Secure sensitive files

chmod 600 /home/ibizaapp/ibizagirl/server/.env

chmod -R 700 /home/ibizaapp/ibizagirl/secure-content

chmod -R 755 /home/ibizaapp/ibizagirl/public

**💳 PayPal Setup**

**Step 1: PayPal Business Account**

1. Go to [PayPal Developer](https://developer.paypal.com)
2. Switch to **Live** environment
3. Create Live App
4. Get Live Client ID and Secret
5. Add them to .env file

**Step 2: Create Subscription Plan**

# Use PayPal API to create subscription plan

curl -X POST https://api.paypal.com/v1/billing/plans \

-H "Authorization: Bearer YOUR\_ACCESS\_TOKEN" \

-H "Content-Type: application/json" \

-d '{

"product\_id": "PROD-XXXXXXXXXXXX",

"name": "IbizaGirl VIP Monthly",

"billing\_cycles": [{

"frequency": {

"interval\_unit": "MONTH",

"interval\_count": 1

},

"tenure\_type": "REGULAR",

"sequence": 1,

"total\_cycles": 0,

"pricing\_scheme": {

"fixed\_price": {

"value": "10",

"currency\_code": "EUR"

}

}

}]

}'

**Step 3: Setup Webhooks**

1. In PayPal Developer Dashboard
2. Add webhook URL: https://ibizagirl.pics/api/webhooks/paypal
3. Subscribe to events:
   * BILLING.SUBSCRIPTION.ACTIVATED
   * BILLING.SUBSCRIPTION.CANCELLED
   * PAYMENT.CAPTURE.COMPLETED

**🌐 CDN Configuration (Cloudflare)**

**Step 1: Add Site to Cloudflare**

1. Sign up at [Cloudflare](https://cloudflare.com)
2. Add your domain
3. Update nameservers at your registrar
4. Wait for propagation (5-30 minutes)

**Step 2: Configure Cloudflare**

SSL/TLS: Full (strict)

Always Use HTTPS: On

Automatic HTTPS Rewrites: On

Minimum TLS Version: 1.2

Caching Level: Standard

Browser Cache TTL: 1 month

Page Rules:

1. \*ibizagirl.pics/api/\* - Cache Level: Bypass

2. \*ibizagirl.pics/secure-content/\* - Access Denied

3. \*ibizagirl.pics/\*.jpg - Cache Everything, Edge Cache TTL: 1 month

**Step 3: Setup Image Optimization**

Polish: Lossy

WebP: On

Mirage: On (for mobile)

**📊 Monitoring & Maintenance**

**Step 1: Setup Monitoring**

# Install monitoring tools

npm install -g clinic

npm install -g 0x

# Setup log rotation

sudo nano /etc/logrotate.d/ibizagirl

/home/ibizaapp/ibizagirl/logs/\*.log {

daily

rotate 14

compress

missingok

notifempty

create 0640 ibizaapp ibizaapp

sharedscripts

postrotate

pm2 reloadLogs

endscript

}

**Step 2: Backup Strategy**

Create backup script:

nano /home/ibizaapp/backup.sh

#!/bin/bash

# Daily backup script

DATE=$(date +%Y%m%d\_%H%M%S)

BACKUP\_DIR="/home/ibizaapp/ibizagirl/backup"

# Backup MongoDB

mongodump --uri="mongodb://ibizaapp:PASSWORD@localhost:27017/ibizagirl" \

--out="$BACKUP\_DIR/mongo\_$DATE"

# Backup files

tar -czf "$BACKUP\_DIR/files\_$DATE.tar.gz" \

/home/ibizaapp/ibizagirl/public \

/home/ibizaapp/ibizagirl/secure-content

# Keep only last 7 days

find $BACKUP\_DIR -type f -mtime +7 -delete

# Optional: Upload to cloud storage

# aws s3 sync $BACKUP\_DIR s3://your-backup-bucket/

# Make executable and add to cron

chmod +x /home/ibizaapp/backup.sh

crontab -e

# Add: 0 3 \* \* \* /home/ibizaapp/backup.sh

**Step 3: Performance Monitoring**

# Monitor with PM2

pm2 monit

# Check logs

pm2 logs

# Monitor system resources

htop

# Check Nginx logs

sudo tail -f /var/log/nginx/ibizagirl\_error.log

**🔧 Troubleshooting**

**Common Issues and Solutions**

**1. 502 Bad Gateway**

# Check if Node app is running

pm2 status

pm2 restart ibizagirl

# Check Node logs

pm2 logs

**2. PayPal Not Working**

# Verify environment variables

cd /home/ibizaapp/ibizagirl/server

cat .env | grep PAYPAL

# Check PayPal webhook logs

tail -f logs/payment.log

**3. Images Not Loading**

# Check file permissions

ls -la /home/ibizaapp/ibizagirl/public/assets/

# Check Nginx error log

sudo tail -f /var/log/nginx/ibizagirl\_error.log

**4. High Memory Usage**

# Restart PM2 with memory limit

pm2 delete all

pm2 start ecosystem.config.js --max-memory-restart 1G

# Clear Redis cache

redis-cli FLUSHDB

**5. SSL Certificate Issues**

# Renew certificate

sudo certbot renew --dry-run

sudo certbot renew

# Restart Nginx

sudo systemctl restart nginx

**🚨 Emergency Procedures**

**Take Site Offline (Maintenance Mode)**

# Create maintenance page

echo "<h1>Under Maintenance</h1>" > /home/ibizaapp/ibizagirl/public/maintenance.html

# Update Nginx to show maintenance

sudo nano /etc/nginx/sites-available/ibizagirl.pics

# Add at the beginning of server block:

# if (-f $document\_root/maintenance.html) {

# return 503;

# }

# error\_page 503 @maintenance;

# location @maintenance {

# rewrite ^(.\*)$ /maintenance.html break;

# }

sudo nginx -t && sudo systemctl reload nginx

**Emergency Rollback**

# Stop current app

pm2 stop all

# Restore from backup

cd /home/ibizaapp

mv ibizagirl ibizagirl\_broken

tar -xzf backup/files\_YYYYMMDD.tar.gz

mongorestore --uri="mongodb://..." backup/mongo\_YYYYMMDD/

# Restart

pm2 start ecosystem.config.js

**📞 Support Contacts**

* **Server Issues**: Your hosting provider support
* **PayPal Issues**: [PayPal Merchant Support](https://www.paypal.com/merchantsupport)
* **Domain/DNS**: Your registrar support
* **SSL Issues**: [Let's Encrypt Community](https://community.letsencrypt.org)

**✅ Post-Deployment Checklist**

* [ ] SSL certificate working (https://ibizagirl.pics)
* [ ] PayPal payments processing correctly
* [ ] All images loading properly
* [ ] Videos playing correctly
* [ ] Mobile responsive working
* [ ] Backup system running
* [ ] Monitoring active
* [ ] Error logging working
* [ ] Rate limiting active
* [ ] Security headers present

**📈 Performance Optimization Tips**

1. **Enable HTTP/2** in Nginx (already in config)
2. **Use WebP images** where possible
3. **Implement Redis caching** for sessions
4. **Use CDN for static assets**
5. **Optimize images** before upload (max 1920px width)
6. **Compress videos** with ffmpeg
7. **Monitor with Google PageSpeed Insights**

**Last Updated**: August 2025 **Version**: 1.0.0 **Author**: IbizaGirl.pics Technical Team