

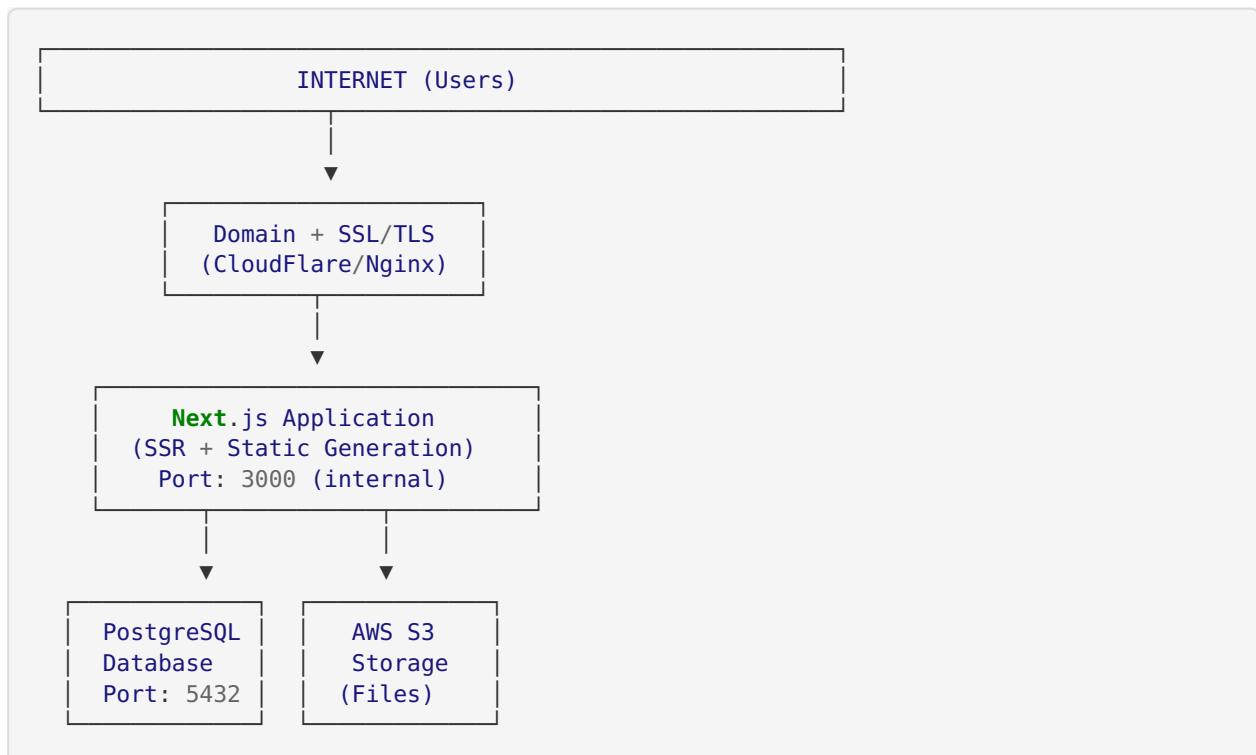
# HƯỚNG DẪN TRIỂN KHAI HỆ THỐNG TẠP CHÍ ĐIỆN TỬ KHOA HỌC HẬU CẦN QUÂN SỰ

## MỤC LỤC

1. Tổng quan hệ thống
2. Yêu cầu hệ thống
3. Chuẩn bị môi trường
4. Cài đặt và cấu hình
5. Cấu hình Database
6. Cấu hình AWS S3
7. Biến môi trường
8. Build và Deploy
9. Cấu hình DNS và Domain
10. Bảo mật và Performance
11. Sao lưu và Khôi phục
12. Monitoring và Logging
13. Troubleshooting
14. Maintenance

## 1. TỔNG QUAN HỆ THỐNG

### 1.1. Kiến trúc hệ thống



## 1.2. Tech Stack

### Frontend:

- Next.js 14.2.28 (React 18.2.0)
- TypeScript 5.2.2
- Tailwind CSS 3.3.3
- Shadcn UI components

### Backend:

- Next.js API Routes
- Prisma ORM 6.7.0
- NextAuth 4.24.11 (Authentication)

### Database:

- PostgreSQL 14+ (với Full-Text Search)

### Storage:

- AWS S3 (File storage)

### Additional Services:

- Cloudflare (CDN + DNS)
- AWS SDK v3

## 1.3. Thư mục dự án

```
/home/ubuntu/tapchi-hcqs/
├── nextjs_space/
│   ├── app/
│   │   ├── (public)/
│   │   ├── api/
│   │   ├── auth/
│   │   ├── dashboard/
│   │   ├── components/
│   │   ├── lib/
│   │   ├── prisma.ts
│   │   ├── auth.ts
│   │   ├── s3.ts
│   │   └── audit-logger.ts
│   ├── prisma/
│   │   └── schema.prisma
│   ├── public/
│   ├── scripts/
│   ├── .env
│   ├── package.json
│   └── next.config.js
└── Uploads/                                # User uploads (demo)

# Next.js application
# App Router (Next.js 13+)
# Public pages
# API routes
# Auth pages
# Protected dashboard
# React components
# Utility libraries
# Prisma client
# Auth helpers
# S3 utilities
# Audit logging
# Database schema
# Static assets
# Utility scripts
# Environment variables
# Dependencies
# Next.js config
```

## 2. YÊU CẦU HỆ THỐNG

### 2.1. Phần cứng (Server Production)

#### Cấu hình tối thiểu:

- **CPU:** 2 cores (x86\_64)
- **RAM:** 4 GB

- **Storage:** 40 GB SSD

- **Network:** 100 Mbps

#### Cấu hình khuyến nghị:

- **CPU:** 4+ cores

- **RAM:** 8+ GB

- **Storage:** 100+ GB SSD (NVMe)

- **Network:** 1 Gbps

## 2.2. Hệ điều hành

- **Ubuntu 20.04 LTS** hoặc **22.04 LTS** (khuyến nghị)
- **Debian 11+**
- **CentOS 8+ / Rocky Linux 8+**

## 2.3. Phần mềm bắt buộc

Phần mềm	Phiên bản tối thiểu	Ghi chú
<b>Node.js</b>	18.17.0+	Khuyến nghị 20.x LTS
<b>Yarn</b>	1.22.0+	Package manager
<b>PostgreSQL</b>	14.0+	Database server
<b>Git</b>	2.30.0+	Version control
<b>Nginx</b>	1.18.0+	Reverse proxy (production)
<b>PM2</b>	5.3.0+	Process manager (khuyến nghị)

## 2.4. Dịch vụ bên ngoài

- **AWS S3:** Lưu trữ file (images, PDFs, videos)
- **Email Service:** SMTP server cho gửi email thông báo
- **Cloudflare** (tùy chọn): CDN + DNS + DDoS protection

## 3. CHUẨN BỊ MÔI TRƯỜNG

### 3.1. Cập nhật hệ thống

```
# Ubuntu/Debian
```

```
sudo apt update && sudo apt upgrade -y
```

```
# CentOS/Rocky Linux
```

```
sudo yum update -y
```

## 3.2. Cài đặt Node.js

### Sử dụng NodeSource (khuyến nghị)

```
# Node.js 20.x LTS
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt-get install -y nodejs

# Kiểm tra phiên bản
node -v # v20.x.x
npm -v # 10.x.x
```

## 3.3. Cài đặt Yarn

```
# Cài Yarn globall
npm install -g yarn

# Kiểm tra
yarn --version # 1.22.x
```

## 3.4. Cài đặt PostgreSQL

### Ubuntu/Debian

```
# Cài PostgreSQL
sudo apt install postgresql postgresql-contrib -y

# Khởi động service
sudo systemctl start postgresql
sudo systemctl enable postgresql

# Kiểm tra
sudo systemctl status postgresql
```

### Tạo database và user

```
# Chuyển sang user postgres
sudo -u postgres psql
```

Trong PostgreSQL shell:

```
-- Tạo user
CREATE USER tapchi_user WITH PASSWORD 'your_strong_password_here';

-- Tạo database
CREATE DATABASE tapchi_hcqs OWNER tapchi_user;

-- Cấp quyền
GRANT ALL PRIVILEGES ON DATABASE tapchi_hcqs TO tapchi_user;

-- Thoát
\q
```

### 3.5. Cài đặt Git

```
sudo apt install git -y
git --version
```

### 3.6. Cài đặt Nginx (cho production)

```
sudo apt install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
```

### 3.7. Cài đặt PM2 (Process Manager)

```
npm install -g pm2
pm2 --version
```

## 4. CÀI ĐẶT VÀ CẤU HÌNH

### 4.1. Clone hoặc upload source code

#### Sử dụng Git (nếu có repository)

```
cd /home/ubuntu
git clone <repository_url> tapchi-hcqs
cd tapchi-hcqs/nextjs_space
```

#### Hoặc upload file ZIP

```
cd /home/ubuntu
unzip tapchi-hcqs.zip
cd tapchi-hcqs/nextjs_space
```

### 4.2. Cài đặt dependencies

```
cd /home/ubuntu/tapchi-hcqs/nextjs_space

# Cài tất cả packages
yarn install

# Kiểm tra không có lỗi
echo $? # Output: 0 (thành công)
```

**Lưu ý:** Quá trình này có thể mất 5-10 phút tùy theo kết nối mạng.

## 4.3. Tạo file .env

```
cd /home/ubuntu/tapchi-hcqs/nextjs_space
cp .env.example .env # Nếu có file mẫu
# Hoặc
nano .env
```

Xem chi tiết các biến môi trường tại [Mục 7](#).

# 5. CẤU HÌNH DATABASE

## 5.1. Cấu hình DATABASE\_URL

Trong file `.env` :

```
DATABASE_URL="postgresql://tapchi_user:your_strong_password_here@localhost:5432/
tapchi_hcqs?schema=public"
```

**Format:**

```
postgresql://[username]:[password]@[host]:[port]/[database]?schema=public
```

## 5.2. Chạy Prisma migrations

```
cd /home/ubuntu/tapchi-hcqs/nextjs_space

# Generate Prisma Client
yarn prisma generate

# Chạy migrations
yarn prisma migrate deploy

# Hoặc nếu là lần đầu setup
yarn prisma db push
```

## 5.3. Thiết lập Full-Text Search (FTS)

```
# Chạy script setup FTS
psql -U tapchi_user -d tapchi_hcqs -f prisma/fts_setup.sql
```

Nội dung `fts_setup.sql` (nếu chưa có):

```
-- Thêm cột search_vector
ALTER TABLE "Submission" ADD COLUMN IF NOT EXISTS search_vector tsvector;

-- Tạo index GIN
CREATE INDEX IF NOT EXISTS submission_search_vector_idx ON "Submission" USING GIN (search_vector);

-- Tạo trigger tự động update
CREATE OR REPLACE FUNCTION submission_search_vector_update() RETURNS trigger AS $$

BEGIN
    NEW.search_vector := 
        setweight(to_tsvector('english', coalesce(NEW.title, '')), 'A') ||
        setweight(to_tsvector('english', coalesce(NEW."abstractVn", '')), 'B') ||
        setweight(to_tsvector('english', coalesce(NEW."abstractEn", '')), 'B') ||
        setweight(to_tsvector('english', coalesce(array_to_string(NEW.keywords, ' '), ''), '')), 'C');
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

DROP TRIGGER IF EXISTS submission_search_vector_trigger ON "Submission";
CREATE TRIGGER submission_search_vector_trigger
BEFORE INSERT OR UPDATE ON "Submission"
FOR EACH ROW EXECUTE FUNCTION submission_search_vector_update();

-- Update dữ liệu cũ
UPDATE "Submission" SET search_vector =
    setweight(to_tsvector('english', coalesce(title, '')), 'A') ||
    setweight(to_tsvector('english', coalesce("abstractVn", '')), 'B') ||
    setweight(to_tsvector('english', coalesce("abstractEn", '')), 'B') ||
    setweight(to_tsvector('english', coalesce(array_to_string(keywords, ' '), ''), '')), 'C')
WHERE search_vector IS NULL;
```

## 5.4. Seed dữ liệu mẫu (tùy chọn)

### Tạo tài khoản test

```
cd /home/ubuntu/tapchi-hcqs/nextjs_space
ts-node reset_and_create_test_users.ts
```

### Danh sách tài khoản test:

Email	Vai trò	Mật khẩu
admin@tapchinckhhcqs.vn	SYSADMIN	TapChi@2025
eic@tapchinckhhcqs.vn	EIC	TapChi@2025
managing.editor@tapchinckhhcqs.vn	MANAGING_EDITOR	TapChi@2025
editor@tapchinckhhcqs.vn	SECTION_EDITOR	TapChi@2025
author@tapchinckhhcqs.vn	AUTHOR	TapChi@2025
reviewer@tapchinckhhcqs.vn	REVIEWER	TapChi@2025
auditor@tapchinckhhcqs.vn	SECURITY_AUDITOR	TapChi@2025

## Seed dữ liệu CMS

```
# Site Settings
ts-node seed_site_settings.ts

# Navigation Menu
ts-node seed_navigation.ts

# Homepage Sections
ts-node seed_homepage_sections.ts

# Public Pages
ts-node seed_public_pages.ts
```

## 5.5. Tạo bảng Media (nếu chưa có)

```
ts-node create_media_table.ts
```

## 5.6. Apply Issue PDF migration

```
ts-node apply_pdf_url_migration.ts
```

# 6. CẤU HÌNH AWS S3

## 6.1. Tạo S3 Bucket

1. Đăng nhập **AWS Console**: <https://console.aws.amazon.com/s3/>
2. Nhấn “**Create bucket**”
3. Điền thông tin:
  - **Bucket name**: tapchi-hcqs-storage (hoặc tên khác)
  - **Region**: ap-southeast-1 (Singapore) hoặc gần nhất

- **Block Public Access:** Bỏ chọn “Block all public access” (cho public files)
- Nhấn “Create bucket”

## 6.2. Cấu hình Bucket Policy (cho public read)

Vào bucket → Permissions → Bucket Policy, thêm:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::tapchi-hcqs-storage/public/*"
    }
  ]
}
```

**Giải thích:** Cho phép public read các file trong thư mục public/

## 6.3. Cấu hình CORS

Vào bucket → Permissions → CORS, thêm:

```
[
  {
    "AllowedHeaders": ["*"],
    "AllowedMethods": ["GET", "PUT", "POST", "DELETE", "HEAD"],
    "AllowedOrigins": [
      "https://tapchinckhhcqs.abacusa1.app",
      "http://localhost:3000"
    ],
    "ExposeHeaders": ["ETag"],
    "MaxAgeSeconds": 3000
  }
]
```

**Lưu ý:** Thay tapchinckhhcqs.abacusa1.app bằng domain thực tế của bạn.

## 6.4. Tạo IAM User cho ứng dụng

1. Vào **IAM Console:** <https://console.aws.amazon.com/iam/>
2. **Users → Add users**
3. **User name:** tapchi-hcqs-app
4. **Access type:**  Programmatic access
5. **Permissions:** Attach policy AmazonS3FullAccess (hoặc tạo policy riêng)
6. **Create user**
7. **Lưu lại:**
  - AWS\_ACCESS\_KEY\_ID
  - AWS\_SECRET\_ACCESS\_KEY

## 6.5. Policy tùy chỉnh (bảo mật hơn)

Thay vì AmazonS3FullAccess, tạo policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3>ListBucket",
        "s3:GetObjectMetadata"
      ],
      "Resource": [
        "arn:aws:s3:::tapchi-hcqs-storage",
        "arn:aws:s3:::tapchi-hcqs-storage/*"
      ]
    }
  ]
}
```

## 6.6. Cấu trúc thư mục S3

```
tapchi-hcqs-storage/
├── 8414/                      # Folder prefix (if used)
│   ├── banners/                # Banner images
│   ├── news/                   # News cover images
│   ├── issues/                 # Issue cover images
│   ├── covers/                 # Issue PDFs
│   ├── pdfs/                   # Author submissions (private)
│   ├── uploads/                # Uploaded videos
│   ├── videos/                 # Media library files
│   └── media/                  # Public accessible files
└── public/
    └── uploads/
```

---

# 7. BIẾN MÔI TRƯỜNG

---

## 7.1. File .env hoàn chỉnh

Tạo file `.env` tại `/home/ubuntu/tapchi-hcqs/nextjs_space/.env`:

```

# =====
# DATABASE
# =====
DATABASE_URL="postgresql://tapchi_user:your_strong_password_here@localhost:5432/
tapchi_hcqs?schema=public"

# =====
# NEXTAUTH (Authentication)
# =====
NEXTAUTH_URL="https://tapchinckhhcq.s.abacusai.app"
NEXTAUTH_SECRET="your-random-secret-key-at-least-32-characters-long"

# JWT Settings
JWT_SECRET="another-random-secret-for-jwt-signing"
JWT_ACCESS_TOKEN_EXPIRY="24h"          # Access token hết hạn sau 24 giờ
JWT_REFRESH_TOKEN_EXPIRY="30d"         # Refresh token hết hạn sau 30 ngày

# =====
# AWS S3 STORAGE
# =====
AWS_REGION="ap-southeast-1"           # Singapore region
AWS_BUCKET_NAME="tapchi-hcqs-storage"  # Tên S3 bucket
AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE" # IAM User Access Key
AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfCYEXAMPLEKEY" # IAM Secret Key
AWS_FOLDER_PREFIX="8414"               # Tùy chọn: prefix cho tất cả file

# =====
# EMAIL (SMTP)
# =====
SMTP_HOST="smtp.gmail.com"            # Hoặc smtp.office365.com
SMTP_PORT="587"                      # 587 cho TLS, 465 cho SSL
SMTP_SECURE="false"                   # false cho TLS, true cho SSL
SMTP_USER="no-reply@tapchinckhhcq.s.vn"
SMTP_PASSWORD="your_email_password_or_app_password"
SMTP_FROM="Tạp chí KHCHHCQS <no-reply@tapchinckhhcq.s.vn>

# =====
# APPLICATION
# =====
NODE_ENV="production"                 # development | production
NEXT_PUBLIC_APP_URL="https://tapchinckhhcq.s.abacusai.app"
NEXT_PUBLIC_API_URL="https://tapchinckhhcq.s.abacusai.app/api"

# =====
# CLOUDFLARE (Optional - for analytics)
# =====
# CLOUDFLARE_ANALYTICS_TOKEN="your.cloudflare_analytics_token"

# =====
# GOOGLE ANALYTICS (Optional)
# =====
# NEXT_PUBLIC_GA_MEASUREMENT_ID="G-XXXXXXXXXX"

# =====
# SECURITY
# =====
# Rate limiting (requests per minute)
RATE_LIMIT_MAX_REQUESTS="100"

# Password policy
PASSWORD_MIN_LENGTH="8"

```

```

# Session settings
SESSION_MAX_AGE="86400"          # 24 hours in seconds

# =====
# FEATURE FLAGS
# =====
ENABLE_REGISTRATION="false"        # Tắt đăng ký công khai
ENABLE_EMAIL_VERIFICATION="true"
ENABLE_TWO_FACTOR_AUTH="false"      # 2FA (chưa implement đầy đủ)

# =====
# LOGGING & MONITORING
# =====
LOG_LEVEL="info"                  # error | warn | info | debug
ENABLE_AUDIT_LOG="true"

# =====
# MAINTENANCE
# =====
MAINTENANCE_MODE="false"          # Bật/tắt chế độ bảo trì

```

## 7.2. Tạo NEXTAUTH\_SECRET

```

# Generate random 32-character string
openssl rand -base64 32

# Hoặc
node -e "console.log(require('crypto').randomBytes(32).toString('base64'))"

```

Sao chép output và dán vào `NEXTAUTH_SECRET` và `JWT_SECRET`.

## 7.3. Bảo mật file .env

```

cd /home/ubuntu/tapchi-hcqs/nextjs_space

# Set permissions (chỉ owner có thể đọc)
chmod 600 .env

# Kiểm tra
ls -la .env
# Output: -rw----- 1 ubuntu ubuntu ...

```

## 8. BUILD VÀ DEPLOY

### 8.1. Build ứng dụng

```
cd /home/ubuntu/tapchi-hcqs/nextjs_space

# Build production
yarn build

# Output sẽ tương tự:
# ✓ Compiled successfully
# ✓ Linting and checking validity of types
# ✓ Collecting page data
# ✓ Generating static pages (xx/xx)
# ✓ Finalizing page optimization
```

**Lưu ý:** Quá trình build có thể mất 5-15 phút.

### 8.2. Test local trước khi deploy

```
# Start production server
yarn start

# Hoặc
NODE_ENV=production node .next/standalone/server.js
```

Truy cập: <http://localhost:3000>

Kiểm tra:

- Trang chủ hiển thị bình thường
- Đăng nhập thành công
- Upload file hoạt động
- Không có lỗi trong Console

Nhấn `Ctrl+C` để dừng.

### 8.3. Deploy với PM2

#### Tạo file ecosystem config

```
cd /home/ubuntu/tapchi-hcqs/nextjs_space
nano ecosystem.config.js
```

Nội dung:

```

module.exports = {
  apps: [
    {
      name: 'tapchi-hcqs',
      script: 'node',
      args: '.next/standalone/server.js',
      cwd: '/home/ubuntu/tapchi-hcqs/nextjs_space',
      instances: 2, // 2 instances cho load balancing
      exec_mode: 'cluster',
      watch: false,
      max_memory_restart: '1G',
      env: {
        NODE_ENV: 'production',
        PORT: 3000
      },
      error_file: './logs/pm2-error.log',
      out_file: './logs/pm2-out.log',
      log_date_format: 'YYYY-MM-DD HH:mm:ss Z',
      merge_logs: true,
      autorestart: true,
      max_restarts: 10,
      min_uptime: '10s'
    }
  ]
};

```

## Start PM2

```

# Tạo thư mục logs
mkdir -p /home/ubuntu/tapchi-hcqs/nextjs_space/logs

# Start app với PM2
pm2 start ecosystem.config.js

# Lưu cấu hình PM2
pm2 save

# Auto-start khi server reboot
pm2 startup
sudo env PATH=$PATH:/usr/bin pm2 startup systemd -u ubuntu --hp /home/ubuntu

# Kiểm tra status
pm2 status
pm2 logs tapchi-hcqs

```

## 8.4. Cấu hình Nginx Reverse Proxy

### Tạo file cấu hình Nginx

```
sudo nano /etc/nginx/sites-available/tapchi-hcqs
```

Nội dung:

```

# Redirect HTTP to HTTPS
server {
    listen 80;
    listen [::]:80;
    server_name tapchinckhhcqs.abacusai.app;

    return 301 https://$server_name$request_uri;
}

# HTTPS Server
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name tapchinckhhcqs.abacusai.app;

    # SSL Certificates (sẽ cấu hình sau với Certbot)
    ssl_certificate /etc/letsencrypt/live/tapchinckhhcqs.abacusai.app/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/tapchinckhhcqs.abacusai.app/privkey.pem;

    # SSL Configuration
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
    ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256';
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 10m;

    # Security Headers
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
    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 Referrer-Policy "no-referrer-when-downgrade" always;

    # Max upload size
    client_max_body_size 100M;

    # Logging
    access_log /var/log/nginx/tapchi-hcqs.access.log;
    error_log /var/log/nginx/tapchi-hcqs.error.log;

    # Gzip Compression
    gzip on;
    gzip_vary on;
    gzip_proxied any;
    gzip_comp_level 6;
    gzip_types text/plain text/css text/xml text/javascript
                application/json application/javascript application/xml+rss
                application/rss+xml font/truetype font/opentype
                application/vnd.ms-fontobject image/svg+xml;

    # Proxy to Next.js app
    location / {
        proxy_pass http://localhost:3000;
        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;
    }
}

```

```

# Timeouts
proxy_connect_timeout 60s;
proxy_send_timeout 60s;
proxy_read_timeout 60s;
}

# Static files caching
location /_next/static {
    proxy_pass http://localhost:3000;
    add_header Cache-Control "public, max-age=31536000, immutable";
}

location /images {
    proxy_pass http://localhost:3000;
    add_header Cache-Control "public, max-age=86400";
}
}

```

## Enable site

```

# Tạo symbolic link
sudo ln -s /etc/nginx/sites-available/tapchi-hcqs /etc/nginx/sites-enabled/

# Test cấu hình
sudo nginx -t

# Nếu OK, reload Nginx
sudo systemctl reload nginx

```

## 8.5. Cài đặt SSL với Let's Encrypt

```

# Cài Certbot
sudo apt install certbot python3-certbot-nginx -y

# Lấy SSL certificate
sudo certbot --nginx -d tapchinckhhcqs.abacusai.app

# Làm theo hướng dẫn, nhập email, đồng ý ToS

# Certbot sẽ tự động:
# - Lấy certificate
# - Cập nhật cấu hình Nginx
# - Setup auto-renewal

# Kiểm tra auto-renewal
sudo certbot renew --dry-run

```

## 8.6. Kiểm tra deployment

```
# Kiểm tra PM2
pm2 status
pm2 logs tapchi-hcqs --lines 50

# Kiểm tra Nginx
sudo systemctl status nginx
sudo tail -f /var/log/nginx/tapchi-hcqs.access.log

# Kiểm tra PostgreSQL
sudo systemctl status postgresql

# Test API health
curl https://tapchinckhhcqs.abacusai.app/api/health
```

Truy cập: <https://tapchinckhhcqs.abacusai.app>

## 9. CẤU HÌNH DNS VÀ DOMAIN

### 9.1. Cấu hình DNS Records (Cloudflare)

1. Đăng nhập **Cloudflare Dashboard**
2. Chọn domain của bạn
3. Vào **DNS → Records**
4. Thêm các record:

Type	Name	Content	Proxy	TTL
A	@	[Server IP]	<input checked="" type="checkbox"/> Proxied	Auto
A	www	[Server IP]	<input checked="" type="checkbox"/> Proxied	Auto
CNAME	tapchinckhhcqs	@	<input checked="" type="checkbox"/> Proxied	Auto

**Lưu ý:** Thay `[Server IP]` bằng IP public của server.

### 9.2. Cấu hình Cloudflare Settings

#### SSL/TLS

- **Mode:** Full (strict)
- **Edge Certificates:** Always Use HTTPS = ON
- **Minimum TLS Version:** 1.2

#### Speed

- **Auto Minify:**  JavaScript, CSS, HTML
- **Brotli:**  ON

#### Caching

- **Caching Level:** Standard
- **Browser Cache TTL:** Respect Existing Headers

## Firewall

- **Security Level:** Medium
- **Bot Fight Mode:**  ON

### 9.3. Test DNS propagation

```
# Kiểm tra DNS
dig tapchinckhhcqs.abacusai.app
nslookup tapchinckhhcqs.abacusai.app

# Hoặc online tool:
# https://dnschecker.org/
```

## 10. BẢO MẬT VÀ PERFORMANCE

### 10.1. Firewall (UFW)

```
# Enable UFW
sudo ufw enable

# Allow SSH (quan trọng!)
sudo ufw allow 22/tcp

# Allow HTTP/HTTPS
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp

# Allow PostgreSQL chỉ từ localhost
sudo ufw allow from 127.0.0.1 to any port 5432

# Kiểm tra
sudo ufw status verbose
```

### 10.2. Fail2Ban (chống brute-force)

```
# Cài Fail2Ban
sudo apt install fail2ban -y

# Tạo local config
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
sudo nano /etc/fail2ban/jail.local
```

Thêm cấu hình cho Nginx:

```
[nginx-limit-req]
enabled = true
filter = nginx-limit-req
action = iptables-multiport[name=ReqLimit, port="http,https", protocol=tcp]
logpath = /var/log/nginx/*error.log
findtime = 600
bantime = 7200
maxretry = 10
```

```
# Restart Fail2Ban
sudo systemctl restart fail2ban
sudo fail2ban-client status
```

### 10.3. PostgreSQL Security

```
sudo nano /etc/postgresql/14/main/postgresql.conf
```

Thay đổi:

```
listen_addresses = 'localhost' # Chỉ cho phép kết nối từ localhost
max_connections = 100          # Giới hạn số connection
shared_buffers = 256MB         # Tăng buffer cho performance
```

```
sudo nano /etc/postgresql/14/main/pg_hba.conf
```

Đảm bảo chỉ có:

local	all	all		peer
host	all	all	127.0.0.1/32	md5
host	all	all	::1/128	md5

```
# Restart PostgreSQL
sudo systemctl restart postgresql
```

### 10.4. Log Rotation

```
sudo nano /etc/logrotate.d/tapchi-hcqs
```

Nội dung:

```
/home/ubuntu/tapchi-hcqs/nextjs_space/logs/*.log {
    daily
    rotate 14
    compress
    delaycompress
    missingok
    notifempty
    create 0640 ubuntu ubuntu
    sharedscripts
    postrotate
        pm2 reloadLogs
    endscript
}
```

## 10.5. Monitoring với PM2

```
# Install PM2 Plus (free tier)
pm2 install pm2-logrotate

# Set logrotate config
pm2 set pm2-logrotate:max_size 50M
pm2 set pm2-logrotate:retain 7
pm2 set pm2-logrotate:compress true

# Monitor
pm2 monit
```

# 11. SAO LƯU VÀ KHÔI PHỤC

## 11.1. Backup Database tự động

### Tạo script backup

```
mkdir -p /home/ubuntu/backups
nano /home/ubuntu/scripts/backup-database.sh
```

Nội dung:

```
#!/bin/bash

# Configuration
DB_NAME="tapchi_hcqs"
DB_USER="tapchi_user"
BACKUP_DIR="/home/ubuntu/backups"
DATE=$(date +"%Y-%m-%d_%H-%M-%S")
BACKUP_FILE="${BACKUP_DIR}/db_${DATE}.sql.gz"
RETENTION_DAYS=30

# Create backup
echo "Starting backup: $BACKUP_FILE"
pg_dump -U $DB_USER -Fc $DB_NAME | gzip > $BACKUP_FILE

# Check if backup was successful
if [ $? -eq 0 ]; then
    echo "Backup completed successfully"

    # Delete old backups
    find $BACKUP_DIR -name "db_*.sql.gz" -mtime +$RETENTION_DAYS -delete
    echo "Old backups deleted (>$RETENTION_DAYS days)"
else
    echo "Backup failed!"
    exit 1
fi
```

```
# Set permissions
chmod +x /home/ubuntu/scripts/backup-database.sh

# Test backup
/home/ubuntu/scripts/backup-database.sh
```

## Cấu hình Cron job

```
crontab -e
```

Thêm dòng (backup mỗi ngày lúc 2:00 AM):

```
0 2 * * * /home/ubuntu/scripts/backup-database.sh >> /home/ubuntu/backups/backup.log
2>&1
```

## 11.2. Backup S3 Files

### Sử dụng AWS CLI

```
# Cài AWS CLI
sudo apt install awscli -y

# Configure
aws configure
# Nhập: Access Key, Secret Key, Region, Output format (json)

# Sync S3 bucket to local
aws s3 sync s3://tapchi-hcqs-storage /home/ubuntu/backups/s3-backup
```

### Script backup S3

```
nano /home/ubuntu/scripts/backup-s3.sh
```

```
#!/bin/bash

BUCKET="tapchi-hcqs-storage"
BACKUP_DIR="/home/ubuntu/backups/s3-backup"
DATE=$(date +"%Y-%m-%d")

echo "Starting S3 sync: $BUCKET -> $BACKUP_DIR/$DATE"
aws s3 sync s3://$BUCKET $BACKUP_DIR/$DATE

if [ $? -eq 0 ]; then
    echo "S3 backup completed"
    # Tar and compress
    tar -czf $BACKUP_DIR/s3_backup_${DATE}.tar.gz -C $BACKUP_DIR/$DATE .
    # Remove uncompressed folder
    rm -rf $BACKUP_DIR/$DATE
else
    echo "S3 backup failed!"
    exit 1
fi
```

```
chmod +x /home/ubuntu/scripts/backup-s3.sh
```

### 11.3. Khôi phục Database

```
# List backups
ls -lh /home/ubuntu/backups/

# Restore từ backup
gunzip -c /home/ubuntu/backups/db_2025-12-08_02-00-00.sql.gz | psql -U tapchi_user -d
tapchi_hcqs

# Hoặc nếu cần drop database trước
sudo -u postgres psql
DROP DATABASE tapchi_hcqs;
CREATE DATABASE tapchi_hcqs OWNER tapchi_user;
\q

gunzip -c /home/ubuntu/backups/db_2025-12-08_02-00-00.sql.gz | psql -U tapchi_user -d
tapchi_hcqs
```

### 11.4. Backup toàn bộ source code

```
# Tạo tarball
tar -czf /home/ubuntu/backups/source_$(date +"%Y%m%d").tar.gz \
--exclude='node_modules' \
--exclude='.next' \
--exclude='logs' \
/home/ubuntu/tapchi-hcqs
```

## 12. MONITORING VÀ LOGGING

### 12.1. PM2 Monitoring

```
# Real-time monitoring
pm2 monit

# View logs
pm2 logs tapchi-hcqs --lines 100
pm2 logs tapchi-hcqs --err --lines 50

# List processes
pm2 list

# Show detailed info
pm2 show tapchi-hcqs
```

## 12.2. System Monitoring

```
# CPU, RAM, Disk usage
htop

# Disk space
df -h

# Network connections
ss -tulpn | grep :3000

# PostgreSQL connections
sudo -u postgres psql -c "SELECT count(*) FROM pg_stat_activity;"
```

## 12.3. Application Logs

### Xem logs trong dashboard

1. Đăng nhập với tài khoản **SYSADMIN** hoặc **SECURITY\_AUDITOR**
2. Vào **Dashboard > Bảo mật > Nhật ký kiểm toán**
3. Lọc theo:
  - Thời gian
  - Loại sự kiện
  - Người dùng

### Xem logs trực tiếp từ database

```
sudo -u postgres psql -d tapchi_hcqs

SELECT * FROM "AuditLog"
WHERE "createdAt" > NOW() - INTERVAL '24 hours'
ORDER BY "createdAt" DESC
LIMIT 100;
```

## 12.4. Nginx Access Logs

```
# Real-time access log
sudo tail -f /var/log/nginx/tapchi-hcqs.access.log

# Error log
sudo tail -f /var/log/nginx/tapchi-hcqs.error.log

# Phân tích top IPs
sudo awk '{print $1}' /var/log/nginx/tapchi-hcqs.access.log | sort | uniq -c | sort -nr | head -20

# Phân tích top URLs
sudo awk '{print $7}' /var/log/nginx/tapchi-hcqs.access.log | sort | uniq -c | sort -nr | head -20
```

## 12.5. Setup Uptime Monitoring

### Sử dụng dịch vụ miễn phí:

- **UptimeRobot**: <https://uptimerobot.com/>
- **Pingdom**: <https://www.pingdom.com/>
- **StatusCake**: <https://www.statuscake.com/>

**Cấu hình:**

1. Tạo tài khoản
2. Thêm monitor:
  - URL: <https://tapchinckhhcqs.abacusai.app>
  - Interval: 5 phút
  - Alert: Email/SMS khi down

## 13. TROUBLESHOOTING

### 13.1. Ứng dụng không khởi động

**Triệu chứng:** pm2 status hiển thị status errored hoặc stopped

**Kiểm tra:**

```
# Xem logs lỗi
pm2 logs tapchi-hcqs --err --lines 50

# Kiểm tra .env file
cat /home/ubuntu/tapchi-hcqs/nextjs_space/.env | grep DATABASE_URL

# Kiểm tra port 3000 có bị chiếm không
sudo lsof -i :3000

# Test kết nối database
psql -U tapchi_user -d tapchi_hcqs -c "SELECT 1;"
```

**Giải pháp:**

```
# Restart app
pm2 restart tapchi-hcqs

# Hoặc delete và start lại
pm2 delete tapchi-hcqs
pm2 start ecosystem.config.js
```

### 13.2. Database connection error

**Lỗi:** PrismaClientInitializationError: Can't reach database server

**Kiểm tra:**

```
# PostgreSQL đang chạy không?
sudo systemctl status postgresql

# Test connection
psql -U tapchi_user -d tapchi_hcqs -h localhost -W

# Kiểm tra pg_hba.conf
sudo cat /etc/postgresql/14/main/pg_hba.conf | grep -v "#"
```

**Giải pháp:**

```
# Restart PostgreSQL
sudo systemctl restart postgresql

# Nếu vẫn lỗi, kiểm tra DATABASE_URL trong .env
# Format phải đúng: postgresql://user:password@host:port/database
```

### 13.3. AWS S3 upload fail

**Lỗi:** S3ServiceException: Access Denied hoặc NoSuchBucket

**Kiểm tra:**

```
# Test AWS credentials
aws s3 ls s3://tapchi-hcqs-storage

# Nếu lỗi, kiểm tra .env
cat .env | grep AWS

# Test upload một file
echo "test" > test.txt
aws s3 cp test.txt s3://tapchi-hcqs-storage/test.txt
```

**Giải pháp:**

1. Kiểm tra IAM User có quyền s3:PutObject , s3:GetObject
2. Kiểm tra bucket name đúng không
3. Kiểm tra region đúng không

### 13.4. 502 Bad Gateway (Nginx)

**Triệu chứng:** Truy cập website hiện lỗi 502

**Kiểm tra:**

```
# App có đang chạy không?
pm2 status
curl http://localhost:3000

# Nginx logs
sudo tail -f /var/log/nginx/error.log
```

**Giải pháp:**

```
# Restart app
pm2 restart tapchi-hcqs

# Restart Nginx
sudo systemctl restart nginx

# Kiểm tra cấu hình Nginx
sudo nginx -t
```

### 13.5. Memory leak / High RAM usage

**Triệu chứng:** App sử dụng RAM cao và tăng dần

**Kiểm tra:**

```
# Xem memory usage
pm2 monit

# Hoặc
free -h
top -p $(pgrep -f 'node.*standalone')
```

**Giải pháp:**

```
# Restart app (giải pháp tạm thời)
pm2 restart tapchi-hcqs

# Giảm số instances trong ecosystem.config.js
# instances: 1 (thay vì 2)

# Set max memory restart
pm2 restart tapchi-hcqs --max-memory-restart 800M
```

**13.6. SSL certificate expired**

Lỗi: ERR\_CERT\_DATE\_INVALID hoặc SSL certificate problem

**Kiểm tra:**

```
# Kiểm tra expiry date
sudo certbot certificates

# Hoặc
openssl s_client -connect tapchinckhhcqs.abacusai.app:443 -servername tapchinckhhc-
qs.abacusai.app | grep "Not After"
```

**Giải pháp:**

```
# Renew certificate manually
sudo certbot renew

# Hoặc force renew
sudo certbot renew --force-renewal

# Reload Nginx
sudo systemctl reload nginx

# Test auto-renewal
sudo certbot renew --dry-run
```

## 14. MAINTENANCE

### 14.1. Update Dependencies

```
cd /home/ubuntu/tapchi-hcqs/nextjs_space

# Kiểm tra outdated packages
yarn outdated

# Update tất cả packages (cẩn thận!)
yarn upgrade

# Hoặc update từng package
yarn upgrade next@latest
yarn upgrade prisma@latest

# Rebuild
yarn build

# Restart
pm2 restart tapchi-hcqs
```

### 14.2. Database Maintenance

#### VACUUM (dọn dẹp database)

```
sudo -u postgres psql -d tapchi_hcqs

-- Vacuum toàn bộ database
VACUUM ANALYZE;

-- Vacuum một bảng cụ thể
VACUUM ANALYZE "Submission";

-- Full vacuum (cần downtime)
VACUUM FULL;
```

#### Reindex

```
sudo -u postgres psql -d tapchi_hcqs

REINDEX DATABASE tapchi_hcqs;
```

### 14.3. Log Cleanup

```
# PM2 logs
pm2 flush # Xóa tất cả logs

# Nginx logs
sudo rm /var/log/nginx/tapchi-hcqs.access.log.*.gz

# System logs
sudo journalctl --vacuum-time=30d # Giữ 30 ngày gần nhất
```

#### 14.4. Security Updates

```
```bash
```

## Ubuntu security updates

---

```
sudo apt update
```

```
sudo apt list -upgradable
```