

Step-by-Step Guide: Pag-host ng Piyukonek sa Railway

Talaan ng Nilalaman

1. [Mga Kailangan](#mga-kailangan)
2. [Pag-setup ng Railway Account](#pag-setup-ng-railway-account)
3. [Pag-handa ng Project](#pag-handa-ng-project)
4. [Pag-deploy sa Railway](#pag-deploy-sa-railway)
5. [Pag-configure ng Database](#pag-configure-ng-database)
6. [Pag-setup ng Environment Variables](#pag-setup-ng-environment-variables)
7. [Pag-test ng Deployment](#pag-test-ng-deployment)
8. [Troubleshooting](#troubleshooting)

1. Mga Kailangan

Bago magsimula, siguraduhing mayroon ka ng:

- ✓ GitHub account (libre)
- ✓ Railway account (libre, may \$5 credit monthly)
- ✓ MySQL database (maaaring gamitin ang Railway MySQL o external provider)
- ✓ Gmail account para sa email functionality
- ✓ Git installed sa computer mo

2. Pag-setup ng Railway Account

Hakbang 2.1: Mag-sign up sa Railway

1. Pumunta sa <https://railway.app>
2. Click ang **Start a New Project** o **Login**
3. Piliin ang **Login with GitHub** (mas madali ito)
4. Pahintulutan ang Railway na ma-access ang GitHub account mo

Hakbang 2.2: Verify ang Account

- Check ang email mo para sa verification link
- Mag-login ulit sa Railway dashboard

3. Pag-handa ng Project

Hakbang 3.1: Siguraduhing Complete ang Files

Tiyakin na mayroon ka ng mga sumusunod na files sa project folder:

```
piyukonekweb/ ■■■■ piyukonek/ ■ ■■■■ app.py ■ ■■■■ templates/ ■ ■■■■ static/ ■■■■  
requirements.txt ■■■■ Procfile ■■■■ .env.example ■■■■ README.md (optional)
```

Hakbang 3.2: I-check ang Procfile

Ang Procfile ay dapat may ganitong content:

```
web: gunicorn -b 0.0.0.0:$PORT piyukonek.app:app
```

Mahalaga:

- Gumagamit ng `\$PORT` variable (automatic ng Railway)
- Tinutukoy ang `piyukonek.app:app` (module path)

Hakbang 3.3: I-check ang requirements.txt

Tiyakin na kasama ang `gunicorn` sa requirements.txt:

```
gunicorn>=21.0.0
```

Hakbang 3.4: I-commit ang Changes sa Git

```
git add . git commit -m "Prepare for Railway deployment" git push origin main
```

Tandaan: Huwag i-commit ang `.env` file! Ito ay para sa local development lang.

4. Pag-deploy sa Railway

Hakbang 4.1: Gumawa ng New Project sa Railway

1. Mag-login sa Railway dashboard

2. Click ang **"New Project"** button
3. Piliin ang **"Deploy from GitHub repo"**
4. Piliin ang repository mo (piyukonekweb)
5. Piliin ang branch (karaniwan ay ``main`` o ``master``)

Hakbang 4.2: I-configure ang Build Settings

1. Sa project dashboard, click ang **"Settings"** tab
2. Sa **"Build Command"**, iwanan itong blank (Railway auto-detect)
3. Sa **"Start Command"**, iwanan itong blank (gagamitin ang Procfile)
4. Sa **"Root Directory"**, iwanan itong blank (kung nasa root ang Procfile)

Hakbang 4.3: I-set ang Python Version (Optional pero Recommended)

1. Sa project root, gumawa ng ``runtime.txt`` file:

`python-3.11.0`

O kaya sa Railway Settings, i-set ang Python version.

Hakbang 4.4: Hintayin ang Build

- Makikita mo ang build logs sa Railway dashboard
- Hintayin hanggang matapos ang build process
- Kapag successful, makikita mo ang **"Deploy Succeeded"** message

5. Pag-configure ng Database

Option A: Gamitin ang Railway MySQL (Recommended para sa Simula)

Hakbang 5.1: Mag-add ng MySQL Service

1. Sa Railway project dashboard, click **"New"**
2. Piliin **"Database"** → **"Add MySQL"**
3. Hintayin ang setup (mga 1-2 minuto)

Hakbang 5.2: Kunin ang Connection String

1. Click ang MySQL service
2. Pumunta sa **"Variables"** tab

3. Hanapin ang `MYSQL_URL` o `DATABASE_URL`
4. Kopyahin ang connection string

Format ng Connection String:

`mysql://user:password@host:port/database`

Hakbang 5.3: I-convert sa PyMySQL Format

Ang Railway MySQL connection string ay kailangan i-convert para sa Flask:

- Original: `mysql://user:pass@host:port/db`
- Converted: `mysql+pymysql://user:pass@host:port/db`

Tandaan: I-add ang `+pymysql` pagkatapos ng `mysql`

Option B: Gamitin ang External MySQL (PlanetScale, AWS RDS, etc.)

1. Kunin ang connection string mula sa provider mo
2. I-convert sa PyMySQL format kung kailangan
3. I-set bilang environment variable (tingnan ang Section 6)

6. Pag-setup ng Environment Variables

Hakbang 6.1: Pumunta sa Variables Tab

1. Sa Railway project dashboard, click ang `Variables` tab
2. O kaya, click ang service name → `Variables`

Hakbang 6.2: I-add ang mga Required Variables

I-add ang mga sumusunod na variables:

Database Configuration

`DATABASE_URL=mysql+pymysql://user:password@host:port/database`

(I-replace ang values mula sa MySQL service mo)

Secret Key

`SECRET_KEY=your-very-long-random-secret-key-here-minimum-32-characters`

Paano gumawa ng Secret Key:

- Sa local terminal: `python -c "import secrets; print(secrets.token_hex(32))"`
- O gumamit ng online generator

- **Mahalaga:** Hindi dapat pareho sa local development!

Email Configuration (Gmail)

```
MAIL_USERNAME=your-email@gmail.com  
MAIL_DEFAULT_SENDER=your-email@gmail.com
```

```
MAIL_PASSWORD=your-app-password
```

Paano gumawa ng Gmail App Password:

1. Pumunta sa Google Account Settings
2. Security → 2-Step Verification (kailangan i-enable muna)
3. App Passwords → Generate
4. Kopyahin ang 16-character password

Production Settings

```
FLASK_ENV=production FLASK_DEBUG=false HTTPS_ENABLED=true SESSION_COOKIE_SECURE=true
```

Optional: API Keys (kung ginagamit)

```
HF_API_TOKEN=your-huggingface-token OPENAI_API_KEY=your-openai-key
```

Hakbang 6.3: I-verify ang Variables

- Tiyakin na lahat ng variables ay naka-set
- Walang typos o extra spaces
- Ang values ay tama

7. Pag-test ng Deployment

Hakbang 7.1: Kunin ang Domain

1. Sa Railway dashboard, pumunta sa **"Settings"** tab
2. Sa **"Domains"** section, makikita mo ang default domain
 - Format: ``your-project-name.up.railway.app``
3. O kaya, mag-generate ng custom domain

Hakbang 7.2: I-test ang Website

1. Buksan ang domain sa browser
2. Tiyakin na:
 - ✓ Naglo-load ang homepage
 - ✓ Walang errors sa console

- ✓ Gumagana ang login/registration
- ✓ Nakakapag-save sa database
- ✓ Gumagana ang email sending (kung may email feature)

Hakbang 7.3: I-check ang Logs

1. Sa Railway dashboard, click ang `Deployments` tab
2. Click ang latest deployment
3. Tingnan ang logs para sa errors o warnings
4. I-check ang `Metrics` para sa performance

8. Troubleshooting

Problem: Build Failed

Mga Posibleng Dahilan:

- ✗ Mali ang Procfile format
- ✗ Kulang ang dependencies sa requirements.txt
- ✗ Mali ang Python version

Solution:

1. I-check ang build logs sa Railway
2. I-verify ang Procfile syntax
3. I-test ang `pip install -r requirements.txt` locally
4. I-update ang runtime.txt kung kailangan

Problem: Application Crashes / 502 Error

Mga Posibleng Dahilan:

- ✗ Mali ang PORT configuration
- ✗ Hindi naka-set ang DATABASE_URL
- ✗ Mali ang module path sa Procfile

Solution:

1. Tiyakin na ang Procfile ay gumagamit ng `$PORT`
2. I-verify ang DATABASE_URL format
3. I-check ang app.py kung tama ang module structure

Problem: Database Connection Error

Mga Posibleng Dahilan:

- ✗ Mali ang DATABASE_URL format
- ✗ Hindi accessible ang database
- ✗ Mali ang credentials

Solution:

1. I-verify ang DATABASE_URL format (dapat may `+pymysql`)
2. I-check kung naka-set ang MySQL service sa Railway
3. I-test ang connection string locally

Problem: Email Not Sending

Mga Posibleng Dahilan:

- ✗ Mali ang Gmail App Password
- ✗ Hindi naka-enable ang 2-Step Verification
- ✗ Mali ang MAIL_USERNAME o MAIL_PASSWORD

Solution:

1. I-verify ang Gmail App Password
2. Tiyakin na naka-enable ang 2-Step Verification
3. I-check ang email variables sa Railway

Problem: Static Files Not Loading

Mga Posibleng Dahilan:

- ✗ Mali ang static folder path
- ✗ Hindi naka-configure ang Flask static folder

Solution:

1. I-verify ang static folder structure
2. I-check ang Flask app configuration
3. Tiyakin na tama ang paths sa templates

9. Best Practices

Security

- ✓ Huwag i-commit ang `.env` file
- ✓ Gumamit ng strong SECRET_KEY

- ✓ I-enable ang HTTPS (automatic sa Railway)
- ✓ I-set ang SESSION_COOKIE_SECURE=true sa production

Performance

- ✓ Gumamit ng gunicorn workers (default ay ok)
- ✓ I-monitor ang memory usage
- ✓ I-optimize ang database queries
- ✓ Gumamit ng CDN para sa static files (optional)

Monitoring

- ✓ Regular na i-check ang Railway logs
- ✓ I-setup ang error tracking (Sentry, etc.)
- ✓ I-monitor ang database usage
- ✓ I-track ang deployment history

10. Additional Resources

Railway Documentation

- Official Docs: <https://docs.railway.app>
- Python Guide: <https://docs.railway.app/guides/python>
- Database Guide: <https://docs.railway.app/databases/mysql>

Support

- Railway Discord: <https://discord.gg/railway>
- Railway Community: <https://community.railway.app>

Useful Commands

```
# Local testing ng gunicorn gunicorn -b 0.0.0.0:5000 piyukonek.app:app # Check ng  
Python version python --version # Install dependencies pip install -r  
requirements.txt
```

Checklist: Bago Mag-deploy

- [] Naka-commit na ang lahat ng changes sa Git
- [] May Procfile na tama ang format
- [] Complete ang requirements.txt (kasama ang unicorn)
- [] Naka-setup na ang Railway account
- [] Naka-connect na ang GitHub repository
- [] Naka-setup na ang MySQL database
- [] Naka-configure na ang lahat ng environment variables
- [] Naka-test na ang application locally
- [] Naka-generate na ang Gmail App Password
- [] Naka-setup na ang SECRET_KEY

Summary ng Quick Steps

1. ****Sign up**** sa Railway (via GitHub)
2. ****Create new project**** → Deploy from GitHub
3. ****Add MySQL**** service
4. ****Set environment variables**** (DATABASE_URL, SECRET_KEY, etc.)
5. ****Wait for deployment**** to complete
6. ****Test**** ang website sa provided domain
7. ****Monitor**** logs at metrics

Good luck sa deployment! ■

Kung may problema, i-check ang Railway logs at documentation.