

AI-04033 Dockerlab to Github

Owner	Ⓜ Rigel Arcayan
Tags	Project
Created time	July 14, 2026 4:46 PM

How to clone a Dockerlab inside my Ubuntu VM hosted in GCP to Github? I have github installed in my Windows11 notebook. Create a step by step instructions.

For your setup (**GCP Ubuntu VM running DockerLab → GitHub backup repository → Windows 11 laptop for management**), the best practice is:

Do NOT clone from Windows first.

Your DockerLab already lives inside Ubuntu on GCP, so the correct workflow is:

```
GCP Ubuntu VM (DockerLab)
    |
    | git init / commit
    ↓
GitHub Repository (Cloud Backup)
    ↑
    |
Windows 11 Laptop (git clone / edit / restore)
```

Your GitHub repository becomes your **Infrastructure-as-Code backup** for:

- docker-compose.yml files
- Dockerfile files
- NGINX Proxy Manager configs
- environment templates
- documentation
- scripts
- Terraform files later

Do **NOT** upload:

- database volumes
 - patient data
 - passwords
 - `.env` files containing secrets
 - SSL certificates
 - backups containing PHI (OpenEMR)
-

PART 1 — Prepare Your GCP Ubuntu DockerLab

Step 1 — SSH into your GCP Ubuntu VM

From Windows PowerShell:

```
ssh username@gcp-external-ip
```

Example:

```
ssh rigel@34.xxx.xxx.xxx
```

Step 2 — Locate your DockerLab directory

Common locations:

```
cd ~  
ls
```

Example:

```
dockerlab
```

Enter:

```
cd dockerlab
```

Verify:

```
ls
```

You should see something like:

```
docker-compose.yml  
openemr/  
openproject/  
docmost/  
npm/  
postgres/  
mariadb/  
portainer/  
scripts/
```

PART 2 — Install Git inside Ubuntu

Check:

```
git --version
```

If missing:

```
sudo apt update  
sudo apt install git -y
```

Verify:

```
git --version
```

Example:

```
git version 2.43.0
```

PART 3 — Configure Git Identity

Set your GitHub name:

```
git config --global user.name "Rigel Arcayan"
```

Set your GitHub email:

```
git config --global user.email "your-github-email@example.com"
```

Verify:

```
git config --list
```

PART 4 — Create Your GitHub Repository

On your Windows 11 browser:

Go to:

[GitHub](#)

Create a new repository.

Example:

Repository name:

GCP-DockerLab-Infrastructure

Recommended:

Private Repository

because your lab contains infrastructure details.

Select:

- ☐ Add README
- ☐ Add .gitignore

Leave empty.

Click:

Create Repository

PART 5 — Prepare DockerLab for Git

Return to Ubuntu:

```
cd ~/dockerlab
```

Initialize Git:

```
git init
```

You should see:

```
Initialized empty Git repository
```

PART 6 — Create .gitignore (IMPORTANT)

Before uploading anything:

nano .gitignore

Add:

```
# Environment secrets
.env
*.env

# Database data
postgres/data/
mariadb/data/
mysql/data/

# Docker volumes
volumes/

# Backups
backup/
backups/

# Logs
*.log

# SSL certificates
*.pem
*.key
*.crt

# OpenEMR patient data
openemr/sites/*/documents/
openemr/sites/*/edi/

# Temporary files
*.tmp
.DS_Store
```

Save:

```
CTRL+O
ENTER
CTRL+X
```

PART 7 — Create Safe Environment Templates

Your real file:

```
.env
```

should NOT go to GitHub.

Create:

```
cp .env .env.example
```

Edit:

```
nano .env.example
```

Replace:

```
PASSWORD=mypassword123
```

with:

```
PASSWORD=CHANGE_ME
```

Example:

```
POSTGRES_USER=openproject  
POSTGRES_PASSWORD=CHANGE_ME  
POSTGRES_DB=openproject
```

Now your repository contains a template only.

PART 8 — First Git Commit

Check what will upload:

```
git status
```

Example:

```
docker-compose.yml  
npm/  
openproject/  
scripts/
```

Add files:

```
git add .
```

Commit:

```
git commit -m "Initial DockerLab infrastructure backup"
```

PART 9 — Connect Ubuntu to GitHub

Copy your GitHub repository URL.

Example:

```
https://github.com/username/GCP-DockerLab-Infrastructure.git
```

Add remote:

```
git remote add origin https://github.com/username/GCP-DockerLab-Infrastructure.git
```

Verify:


```
git remote -v
```

You should see:

```
origin https://github.com/username/GCP-DockerLab-Infrastructure.git
```

PART 10 — Push DockerLab to GitHub

Rename branch:

```
git branch -M main
```

Push:

```
git push -u origin main
```

GitHub will ask authentication.

Use:

- GitHub username
- Personal Access Token (PAT)

NOT your password.

PART 11 — Verify on GitHub

Open:

```
Your Repository
```

You should see:

```
docker-compose.yml
openemr/
openproject/
docmost/
npm/
scripts/
README.md
```

PART 12 — Clone Your DockerLab to Windows 11

Now your Windows laptop becomes your management workstation.

Open:

```
PowerShell
```

Go to your workspace:

```
cd C:\Users\YourName\Documents
```

Clone:

```
git clone https://github.com/username/GCP-DockerLab-Infrastructure.git
```

Example:

```
C:\Users\Rigel\Documents\
├──
└── GCP-DockerLab-Infrastructure
```

Enter:

```
cd GCP-DockerLab-Infrastructure
```

Verify:

```
dir
```

PART 13 — Future Backup Workflow

Whenever you change DockerLab:

Example:

```
nano docker-compose.yml
```

Run:

```
git status
```

Review:

```
git diff
```

Save:

```
git add .
```

Commit:

```
git commit -m "Updated OpenProject database configuration"
```

Upload:

```
git push
```

PART 14 — Recommended DockerLab Git Structure

For your environment, I recommend this:

```
GCP-DockerLab-Infrastructure
├── README.md
├── docker-compose/
│   ├── openemr/
│   │   └── docker-compose.yml
│   ├── openproject/
│   │   └── docker-compose.yml
│   ├── docmost/
│   ├── npm/
│   └── portainer/
├── database/
│   ├── postgres/
│   └── mariadb/
├── scripts/
│   ├── backup.sh
│   └── restore.sh
├── docs/
│   ├── architecture.md
│   └── disaster-recovery.md
└── .gitignore
```

Your Next DevOps Learning Step

For your current DockerLab, the natural progression is:

```
GitHub
  ↓
Docker Compose Management
  ↓
Secrets Management (.env → GitHub Secrets)
  ↓
Automated Backups
  ↓
GitHub Actions CI/CD
  ↓
Terraform GCP Infrastructure
  ↓
Kubernetes
```

Your DockerLab is now becoming a real **Infrastructure-as-Code laboratory**. The next major milestone after GitHub backup would be creating a **complete disaster recovery workflow: "destroy GCP VM → rebuild DockerLab from GitHub → restore databases."**