

# AI-03919 Google Cloud Wordpress Docker

Owner	Ⓜ Rigel Arcayan
Tags	Project
Created time	June 10, 2026 8:15 PM

**Create a step by step guide to create a Wordpress Site (docker) in GCP from signing up for a GCP Account, selecting Free tier Ubuntu VM capable setup until login at Wordpress site and creating the Admin Wordpress site.**

Setting up a self-hosted WordPress site inside a Docker container on Google Cloud Platform (GCP) is an excellent way to utilize GCP's **Always Free Tier**. By sticking to specific constraints, you can run this setup with zero infrastructure costs.

Here is the complete, step-by-step technical guide to executing this deployment from scratch.

## Step 1: Sign Up for a GCP Account & Set Up Billing

To prevent accidental charges, you must configure your account carefully. GCP requires a credit card for identity verification, but you will not be charged if you stay within the free tier boundaries.

1. Go to the [Google Cloud Console](#).
2. Sign in with your Google account. If you do not have one, create it.
3. Complete the registration form, accept the Terms of Service, and enter your payment verification details.
4. Once registered, you will receive a **\$300 free trial credit** valid for 90 days.
5. Create a new project by clicking the project dropdown menu in the top-left header and selecting **New Project**. Name it something distinct (e.g., `wordpress-free-project`).

⚠️ **Critical Cost Prevention:** GCP does not offer an automated billing hard-cap. Go to **Billing > Budgets & Alerts** in the left menu, and set up a monthly budget alert of **\$1.00** to notify you instantly via email if any resource breaches the free tier limits.

## Step 2: Provision the Free-Tier Ubuntu VM

To ensure your virtual machine falls strictly under GCP's **Always Free Tier**, you must select precise configuration options. Selecting an incorrect region, disk type, or machine family will result in consumption charges against your credits or card.

1. In the left navigation menu, navigate to **Compute Engine > VM Instances**.  
Enable the Compute Engine API if prompted.
2. Click **Create Instance**.
3. Configure the VM parameters **exactly** as follows:
  - **Name:** `wordpress-docker-vm`
  - **Region:** Choose **one** of the following three eligible US regions:
    - `us-west1` (Oregon)
    - `us-centrall1` (Iowa)
    - `us-east1` (South Carolina)
  - **Zone:** Any zone within your selected region.
  - **Machine Configuration: \* Series:** `E2`
    - **Machine Type:** `e2-micro` (2 vCPUs, 1 GB memory).
4. Scroll down to **Boot Disk** and click **Change**:
  - **Operating System:** `Ubuntu`
  - **Version:** `Ubuntu 24.04 LTS` (or latest stable LTS)
  - **Boot Disk Type:** Select **Standard Persistent Disk** (Do *not* choose Balanced or SSD).
  - **Size:** `30` GB (This is the maximum free allowance).
  - Click **Select**.

5. Scroll down to **Firewall** and check both:

- `Allow HTTP traffic`
- `Allow HTTPS traffic`

6. Click **Create** at the bottom of the page. Wait 1–2 minutes for the instance to provision and display a green checkmark next to its name. Note your VM's **External IP** address.

## Step 3: Install Docker and Docker Compose on Ubuntu

With the VM active, connect to its terminal to install the container runtime framework.

1. In your VM Instances list, look for your VM and click the **SSH** button in the "Connect" column. This opens an in-browser secure shell terminal.
2. Update the local package index and upgrade existing system packages: Bash

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

3. Install Docker's required dependencies: Bash

```
sudo apt install -y apt-transport-https ca-certificates curl software-properties-common
```

4. Add Docker's official GPG key and stable repository: Bash

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.p/docker.list > /dev/null
```

5. Install Docker Engine and the Docker Compose plugin: Bash

```
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
-compose-plugin
```

6. Verify the installation by checking the status of the Docker service: Bash

```
sudo systemctl status docker --no-pager
```

## Step 4: Configure and Launch WordPress via Docker Compose

Using a single orchestrator file keeps the configuration highly isolated and maintainable, mapping persistent database storage straight to your free 30 GB standard persistent disk.

1. Create a dedicated project directory and navigate into it: Bash

```
mkdir ~/wordpress-site && cd ~/wordpress-site
```

2. Create and open a configuration file named `docker-compose.yml`: Bash

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

3. Paste the following configuration content into the editor. **Change the placeholder passwords** to secure, unique credentials: YAML

```
services:
  db:
    image: mysql:8.0
    container_name: wordpress_db
```

```
restart: always
environment:
  MYSQL_ROOT_PASSWORD: secure_root_password_here
  MYSQL_DATABASE: wordpress
  MYSQL_USER: wp_user
  MYSQL_PASSWORD: secure_db_password_here
volumes:
  - db_data:/var/lib/mysql

wordpress:
  depends_on:
    - db
  image: wordpress:latest
  container_name: wordpress_app
  restart: always
  ports:
    - "80:80"
  environment:
    WORDPRESS_DB_HOST: db:3306
    WORDPRESS_DB_USER: wp_user
    WORDPRESS_DB_PASSWORD: secure_db_password_here
    WORDPRESS_DB_NAME: wordpress
  volumes:
    - wp_data:/var/www/html

volumes:
  db_data:
  wp_data:
```

4. Press `Ctrl + O`, then `Enter` to save the file. Press `Ctrl + X` to exit the nano editor.
5. Launch the containers in detached (background) mode: Bash

```
sudo docker compose up -d
```

6. Confirm that both containers are running actively: Bash

```
sudo docker ps
```

## Step 5: Initialize the WordPress Administrative Setup

Your applications are now successfully serving web traffic from inside the Docker virtual layer.

1. Open a web browser on your computer and type the **External IP** address of your GCP VM into the address bar: Plaintext

```
http://[YOUR_GCP_VM_EXTERNAL_IP]
```

2. The WordPress initialization screen will load. Select your preferred language and click **Continue**.
3. Fill out the administrative site deployment fields:
  - **Site Title:** Enter your website's name.
  - **Username:** Choose a secure administrative username (avoid using "admin").
  - **Password:** Record the secure auto-generated password or create a new one.
  - **Your Email:** Input your functional email address for recovery notifications.
4. Click **Install WordPress**.
5. Once deployment completes, click **Log In**.
6. Provide your administrative credentials on the subsequent `/wp-login.php` screen.

You will be redirected straight to the **WordPress Dashboard**, confirming that your containerized architecture is fully operational on Google Cloud Platform's free tier.