

# Panduan Deploy Dengan Cloud Run

---

## Tutorial Deploy Simple - Step by Step

### Step 1: Buat Project

- Buka [console.cloud.google.com](https://console.cloud.google.com)
- Klik dropdown project di atas
- Klik **"New Project"**
- Isi nama project
- Klik **"Create"**

### Step 2: Enable API Registry

- Di search bar, ketik **"Artifact Registry"**
- Klik **"Artifact Registry API"**
- Klik **"Enable"**
- Tunggu sampai selesai

### Step 3: Buat Repo Registry

No repositories to display  
Create repositories to store, manage, and secure your build artifacts.

[Create repository](#) [Learn more](#)

- Masih di Artifact Registry, klik **" + CREATE REPOSITORY "**
- Isi form:
  - Name:** my-repo
  - Format:** Docker
  - Location:** asia-southeast2 (Jakarta)
- Klik **"CREATE"**

### Step 4: Buka Cloud Shell

- Klik icon Cloud Shell (>) di pojok kanan atas
- Tunggu sampai terminal muncul

### Step 5: Clone Repo

git clone https://github.com/username/your-repo-name.git

### Step 6: Masuk ke Root Aplikasi

```
meutiad6@cloudshell:~ (run-demo-461110)$ git clone https://github.com/Ilhamnofaldi/demo
fatal: destination path 'demogcr' already exists and is not an empty directory.
meutiad6@cloudshell:~ (run-demo-461110)$ cd demogcr
meutiad6@cloudshell:~/demogcr (run-demo-461110)$
```

cd nama-folder-repo

ls

Pastikan terlihat file *package.json*, *Dockerfile*, dll

### Step 7: Cek Dockerfile

cat Dockerfile

Pastikan Dockerfile sudah ada dengan konfigurasi yang benar

### Step 8: Buka Editor

- Klik "Open Editor" di Cloud Shell
- Buka terminal di editor (Terminal > New Terminal)
- Pastikan sudah di path folder yang benar:

pwd

ls

### Step 9: Edit Vite Config (jika menggunakan Vite)

nano vite.config.ts

Tambahkan:

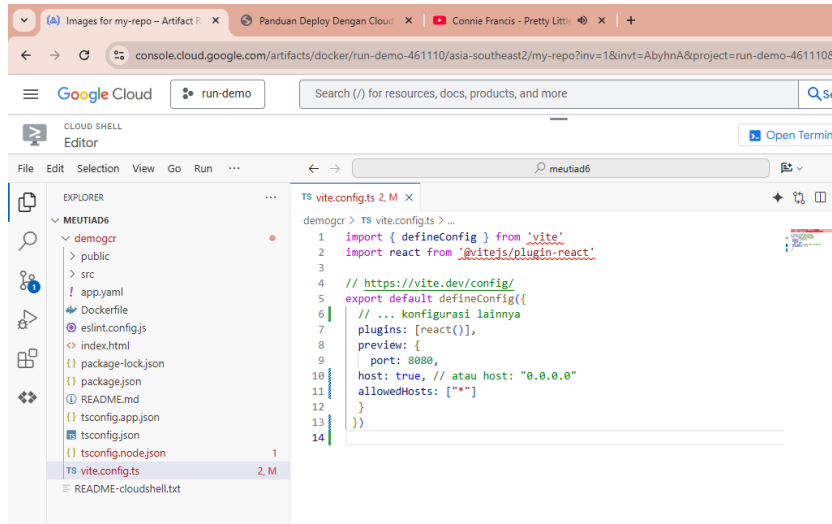
```
export default defineConfig({
  // ... konfigurasi lainnya
  server: {
    host: true, // atau host: "0.0.0.0"
```

```

    allowedHosts: ["*"]
  }
}

```

Tekan Ctrl+X, Y, Enter untuk save



## Step 10: Setup Project

gcloud config set project PROJECT\_ID

Ganti PROJECT\_ID dengan ID project Anda

```

meutiad6@cloudshell:~ (run-demo-461110) $ gcloud config set project run-demo-461110
Updated property [core/project].
meutiad6@cloudshell:~ (run-demo-461110) $

```

## Step 11: Authentication

gcloud auth configure-docker asia-southeast2-docker.pkg.dev

## Step 12: Set Variables

```

export PROJECT_ID=$(gcloud config get-value project)
export IMAGE_NAME="my-app"
export REPO_URL="asia-southeast2-docker.pkg.dev/$PROJECT_ID/my-repo/$IMAGE_NAME"

```

```
meutiad6@cloudshell:~ (run-demo-461110) $ export PROJECT_ID=$(gcloud config get-value project)
export IMAGE_NAME="my-app"
export REPO_URL="asia-southeast2-docker.pkg.dev/$PROJECT_ID/my-repo/$IMAGE_NAME"
```

## Step 13: Build Image

`docker build -t $IMAGE_NAME .`

```
meutiad6@cloudshell:~/demo:~$ docker build -t $IMAGE_NAME .
[*] Building 57.9s (11/11) FINISHED
-> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/node:22
=> [internal] load .dockerignore
=> transferring context: 2B
=> [1/6] FROM docker.io/library/node:22@sha256:0b5b940c21ab03353de9042f9166c75b0cf053c4cd0508c7fd88576646adb7575
=> resolve docker.io/library/node:22@sha256:0b5b940c21ab03353de9042f9166c75b0cf053c4cd0508c7fd88576646adb7575
=> sha256:c3893ebab274d36d6f28d24dc6609e1af4f2829c66a1f01c784ef683cad85b 2.49kB / 2.49kB
=> sha256:3980e114e19722424e4898ed1810c78b2108adcd053c20294964 48.49MB / 48.49MB
=> sha256:0b5b940c21ab03353de9042f9166c75b0cf053c4cd0508c7fd88576646adb7575 6.41kB / 6.41kB
=> sha256:37927ed901b1b2608b72796c6881b7645480268eca4ac9a37b9219e050bb4d84 24.02MB / 24.02MB
```

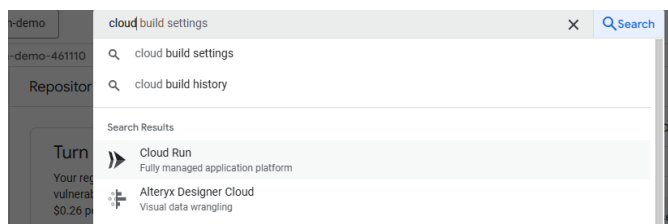
## Step 14: Tag Image

`docker tag $IMAGE_NAME $REPO_URL:latest`

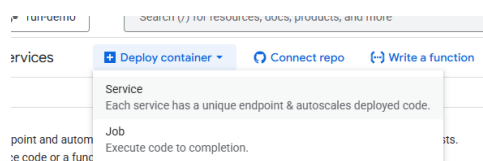
## Step 15: Push Image

`docker push $REPO_URL:latest`

## Step 16: Deploy ke Cloud Run via GCP Console



- Buka Cloud Run di GCP Console
- Klik "+ CREATE SERVICE"



**Container image URL:** Copy dari terminal:

`echo $REPO_URL:latest`

Container image URL Select

[Test with a sample container](#)

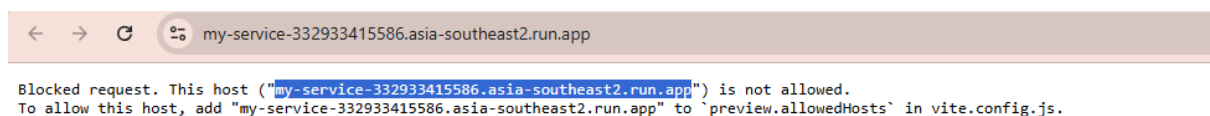
Should listen for HTTP requests on \$PORT (default: 8080) and not rely on local state. [How to build a container?](#)

- **Service name:** my-service
- **Region:** asia-southeast2 (Jakarta)
- **CPU allocation:** CPU is only allocated during request processing
- **Authentication:** Allow unauthenticated invocations
- **Container port:** 8080
- **Klik "CREATE"**

The screenshot shows the Google Cloud Run console. At the top, there's a search bar and navigation tabs for 'Cloud Run', 'Service details', 'Edit & deploy new revision', and 'Set up Continuous Deployment'. A status bar indicates 'Creating service' with a progress indicator. Below this, a list of steps shows 'Creating service', 'Setting IAM policy', 'Creating revision', and 'Routing traffic' all as 'Completed'. The main section displays 'my-service' with its region 'asia-southeast2' and URL 'https://my-service-332933415586.asia-southeast2.run.app'. It also shows 'Scaling: Auto (Min: 0)'. A 'Revisions' table lists one revision: 'my-service-00001-9zx' with 100% traffic and deployed 1 minute ago. To the right, a sidebar shows details for the selected revision, including 'Containers', 'Volumes', 'Networking', 'Security', and 'YAML'. The 'General' tab is active, showing 'Billing' as 'Request-based' and 'Startup CPU boost' as 'Enabled'. At the bottom, a 'CLOUD SHELL' terminal is open, showing the URL 'https://my-service-332933415586.asia-southeast2.run.app'.

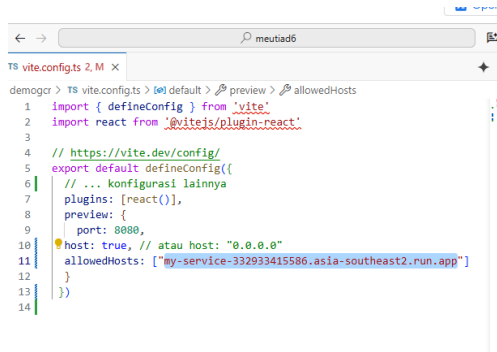
## Step 17: Test

- Setelah deploy selesai, akan muncul URL
- Buka URL di browser untuk test aplikasi
- Akan muncul seperti ini



## Step 18: Ubah AllowedHosts

- Copy url tanpa https
- Lalu save



```
1 import { defineConfig } from 'vite'
2 import react from '@vitejs/plugin-react'
3
4 // https://vite.dev/config/
5 export default defineConfig({
6   // ... konfigurasi lainnya
7   plugins: [react()],
8   preview: {
9     port: 8080,
10    host: true, // atau host: "0.0.0.0"
11    allowedHosts: ["my-service-332933415586.asia-southeast2.run.app"]
12  }
13 })
14
```

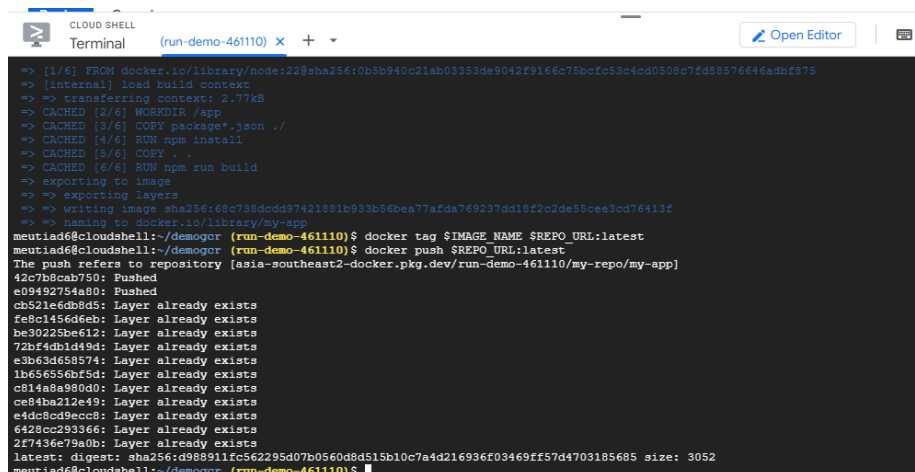
## Step 19: Build ulang

```
export PROJECT_ID=$(gcloud config get-value project)
export IMAGE_NAME="my-app"
export REPO_URL="asia-southeast2-docker.pkg.dev/$PROJECT_ID/my-repo/$IMAGE_NAME"
```

```
docker build -t $IMAGE_NAME .
```

```
docker tag $IMAGE_NAME $REPO_URL:latest
```

```
docker push $REPO_URL:latest
```



```
>> [1/6] FROM docker.io/library/node:22@sha256:0b5b940c21ab03353de9042f9166c75bcfc53c4cd0508c7fd8857664eadbf375
>> [internal] load build context
>> transferring context: 2.77kB
>> CACHED [2/6] WORKDIR /app
>> CACHED [3/6] COPY package*.json ./
>> CACHED [4/6] RUN npm install
>> CACHED [5/6] COPY .
>> CACHED [6/6] RUN npm run build
>> exporting to image
>> exporting layers
>> writing image sha256:68c738dcd97421881b933b56bea77afda769237dd18f2c2de55cee3cd76413f
>> naming to docker.io/library/my-app
meutiad6@cloudshell:~/demogor (run-demo-461110)$ docker tag $IMAGE_NAME $REPO_URL:latest
meutiad6@cloudshell:~/demogor (run-demo-461110)$ docker push $REPO_URL:latest
The push refers to repository [asia-southeast2-docker.pkg.dev/run-demo-461110/my-repo/my-app]
42c7b3cab750: Pushed
e09492754a80: Pushed
cb521e6db8d5: Layer already exists
fe8c1456d6eb: Layer already exists
be30225be612: Layer already exists
72b4f4b1d49d: Layer already exists
e3b634658574: Layer already exists
1b656356bf5d: Layer already exists
c814a8a980d0: Layer already exists
ce84ba212e49: Layer already exists
e4dc8cd9ecc8: Layer already exists
6428cc293366: Layer already exists
2f7436e79a0b: Layer already exists
latest: digest: sha256:6d989911fc562295d07b0560d8d515b10c7a4d216936f03469ff57d4703185685 size: 3052
meutiad6@cloudshell:~/demogor (run-demo-461110)$
```

## Step 20: Ganti dengan container image baru

- Buka cloud run
- Pilih **my-service**
- Lalu pilih **Edit & deploy new revision**
- Pilih select Container image URL

- Lalu ganti container image yang terbaru

## Step 21: Refresh ulang

- Refresh ulang website

## Commands Lengkap (Copy-Paste)

### # Setup project

```
export PROJECT_ID=your-project-id-here
gcloud config set project $PROJECT_ID
gcloud auth configure-docker asia-southeast2-docker.pkg.dev
```

### # Build dan push

```
export IMAGE_NAME="my-app"
export REPO_URL="asia-southeast2-docker.pkg.dev/$PROJECT_ID/my-repo/$IMAGE_NAME"
docker build -t $IMAGE_NAME .
docker tag $IMAGE_NAME $REPO_URL:latest
docker push $REPO_URL:latest
```

### # Deploy via GCP Console

```
echo "Image URL untuk deploy:"
echo $REPO_URL:latest
```

### Untuk Deploy:

- Buka Cloud Run di GCP Console
- CREATE SERVICE
- Paste image URL dari terminal
- Region: Jakarta (asia-southeast2)

- Allow unauthenticated invocations