

# ✓ Deploy Angular 19 to Vercel (Web App Guide)

## 1 Create a Vercel Account

Go to <https://vercel.com>

Sign up with **GitHub**, **GitLab**, or **Bitbucket**

Authorize Vercel to access your repos.

## 2 Push Your Code to Git

Make sure your project is pushed to **GitHub**, **GitLab**, or **Bitbucket**:

```
git init
git remote add origin https://github.com/yourusername/your-repo.git
git add .
git commit -m "Initial commit"
git push -u origin main
```

## 3 Connect Your Repository to Vercel

Click “**New Project**” in your Vercel dashboard.

Select your Git provider (e.g., GitHub).

Find and **import** your Angular repo.

## 4 Configure Build Settings

Vercel may auto-detect Angular and use `npm run build`.

If not, configure manually:

- **Framework Preset:** Other
- **Build Command:** `npm run build`
- **Output Directory:** `dist/<project-name>`

Replace `<project-name>` with your actual Angular app name in `angular.json`.

```
"projects": {
  "mdb": {
```

```

    ...
  }
}

```

👉 Output folder = `dist/mdb`

## 5 Add Environment Variables (if needed)

If you use a TMDB API key:

Go to **Environment Variables** → Add:

- `TMDB_API_KEY` → `YOUR_TMDB_API_KEY`

Vercel will inject this during build.

## 6 Deploy

Click **Deploy**.

Vercel installs dependencies ( `npm install` ), runs `npm run build` , and deploys your app.

## 7 Check Your Live URL

After the build, Vercel gives you a live URL, e.g., `https://your-app.vercel.app` .

You can set up a custom domain if you like.

## 8 Automatic Deploys

Any push to your Git branch (e.g., `main` ) triggers an automatic redeploy.

No extra CI/CD script needed!



## Tips

- Test `ng build` locally before pushing.
- Check `outputPath` in `angular.json` if the build output is wrong.
- Add a `vercel.json` if you need rewrites or redirects:

```

{
  "rewrites": [{ "source": "/(.*)", "destination": "/" } ]
}

```



## Done!

Your Angular 19 app is now deployed on Vercel with auto deploys on every push — no CLI needed.