

React + Vite Deployment Documentation

Deploying React Vite Application on Netlify

1. Introduction

Netlify is a popular cloud platform used to deploy frontend applications quickly with continuous deployment from GitHub.

This documentation explains **step-by-step deployment of a React application created using Vite** on Netlify.

2. Prerequisites

Before deployment, ensure you have:

- Node.js installed (v16+ recommended)
 - Basic knowledge of React
 - A React project created using Vite
 - A GitHub account
 - A Netlify account
-

3. Create React App Using Vite (If Not Created)

```
npm create vite@latest my-react-app  
cd my-react-app  
npm install  
npm run dev
```

Choose:

- Framework: React
 - Variant: JavaScript or TypeScript
-

4. Project Structure

```
my-react-app/  
├── index.html  
├── package.json  
├── vite.config.js  
├── public/  
└── src/  
    └── main.jsx
```

```
├─ App.jsx
├─ components/
```

5. package.json

```
{
  "name": "react-vite-app",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0"
  },
  "devDependencies": {
    "@vitejs/plugin-react": "^4.0.0",
    "vite": "^4.0.0"
  }
}
```

6. Vite Configuration (vite.config.js)

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

export default defineConfig({
  plugins: [react()],
  base: '/'
})
```

7. Build React App for Production

Run the following command:

```
npm run build
```

This will generate a **dist/** folder containing production-ready files.

8. Push Project to GitHub

```
git init
git add .
git commit -m "Initial React Vite commit"
git branch -M main
git remote add origin https://github.com/username/repo-name.git
git push -u origin main
```

9. Deploy on Netlify (Recommended – GitHub)

Step 1: Login to Netlify

- Open Netlify Dashboard
- Click **Add new site** → **Import an existing project**

Step 2: Connect GitHub Repository

- Choose GitHub
- Select your React Vite repository

Step 3: Build Settings

Setting	Value
Build Command	npm run build
Publish Dir	dist

Click **Deploy Site**

10. Environment Variables (If Using API)

If your React app uses backend APIs:

1. Go to **Site Settings** → **Environment Variables**
2. Add variables:

```
VITE_API_URL=https://your-backend-url.com
```

Access in code using:

```
import.meta.env.VITE_API_URL
```

11. Netlify Redirects (For React Routing)

If using React Router, create a file:

```
public/_redirects
```

Add:

```
/*    /index.html    200
```

This prevents 404 errors on page refresh.

12. Manual Deployment (Drag & Drop)

Alternative method:

1. Run `npm run build`
 2. Go to Netlify Dashboard
 3. Drag and drop the **dist/** folder
-

13. Verify Deployment

- Netlify provides a live URL:

```
https://your-site-name.netlify.app
```

- Test all routes
 - Test API integration
-

14. Common Issues & Fixes

Blank page after deployment

- Ensure Publish Directory is `dist`
- Check build command

API not working

- Confirm environment variables
- Restart deployment after changes

Page refresh gives 404

- Add `_redirects` file
-

15. Best Practices

- Use environment variables for API URLs
 - Keep `.env` file out of GitHub
 - Always test `npm run build` locally
 - Use Netlify auto-deploy on push
-

16. Use Cases

- React frontend for Express / Flask backend
 - Student & internship projects
 - Production-ready frontend hosting
-

17. Conclusion

Netlify combined with Vite provides a fast, reliable, and beginner-friendly solution for deploying React applications. With continuous deployment and minimal configuration, your frontend can go live within minutes.