# Deploying a Streamlit Application Using Streamlit Cloud

## Introduction

This document provides a comprehensive guide on deploying a Streamlit application using Streamlit Cloud, ensuring 24/7 accessibility for public users.

## 1. What is Streamlit Cloud?

Streamlit Cloud is a hosting platform specifically designed for Streamlit applications. It allows users to deploy apps directly from GitHub repositories with minimal setup.

### **Why Use Streamlit Cloud?**

✅ Free hosting for public repositories.  
✅ No server setup required – fully managed.  
✅ Easy deployment via GitHub integration.  
✅ 24/7 accessibility (as long as your repo remains public and app resources are within limits).

### **Limitations of Free Plan**

❌ Limited system resources (RAM & CPU).  
❌ No built-in database storage.  
❌ Requires GitHub integration.  
❌ May experience timeouts if heavy processing is involved.

## 2. Prerequisites

Before deploying, ensure you have:

* A GitHub account (to store your Streamlit app code).
* A working Streamlit app (e.g., app.py).
* A requirements file (requirements.txt) specifying dependencies.

## 3. Steps to Deploy on Streamlit Cloud

### **Step 1: Prepare Your Code**

Ensure your Streamlit app is structured as follows:

📂 MyStreamlitApp/

│-- app.py # Main Streamlit application

│-- requirements.txt # List of dependencies (e.g., pandas, numpy, etc.)

│-- README.md # Project description (optional but recommended)

Example requirements.txt:

streamlit

pandas

numpy

matplotlib

### **Step 2: Push Code to GitHub**

1. Create a new GitHub repository.
2. Upload your app.py and requirements.txt files.
3. Commit and push the changes.

### **Step 3: Deploy on Streamlit Cloud**

1. Go to [Streamlit Cloud](https://share.streamlit.io/).
2. Click "Sign in with GitHub".
3. Click "New App".
4. Select your GitHub repository and branch.
5. Specify app.py as the entry point.
6. Click "Deploy".

### **Step 4: Get Public URL**

Once deployed, Streamlit Cloud provides a public URL, such as:

https://your-app-name.streamlit.app

Share this link with users to access your app 24/7.

## 4. Accessibility and Uptime

| Feature | Availability |
| --- | --- |
| Public Access | ✅ 24/7 for public GitHub repos |
| Free Hosting | ✅ Yes |
| Time Limit | ❌ No timeouts unless resource limits are exceeded |
| Data Storage | ❌ No built-in storage (use external DB) |
| Resource Constraints | ✅ Yes (limited RAM & CPU) |

## 5. Best Practices for High Availability

✅ Use optimized code to reduce memory usage.  
✅ Use an external database (e.g., Firebase, PostgreSQL) for persistent data storage.  
✅ Monitor app performance to avoid exceeding resource limits.  
✅ Upgrade to paid plan if more computational power is needed.

## 6. Alternative Hosting Options

If Streamlit Cloud's limitations affect your use case, consider:

| Alternative | Best For |
| --- | --- |
| Railway.app | Small web applications |
| Render.com | Free web services with auto-deployment |
| Heroku | Flexible cloud deployment |
| AWS EC2 | High-performance & scalable apps |

## 7. Conclusion

Streamlit Cloud is an excellent choice for deploying Streamlit apps quickly and for free. However, for long-term, high-performance applications, consider paid hosting options or cloud providers like AWS, Render, or Railway.

✅ Best for: Quick sharing, testing, and small-scale apps.  
❌ Not ideal for: Heavy applications requiring high processing power.

Need help? Let me know if you need further modifications!