# Deploying a Streamlit Application Using Ngrok

## Introduction

This document outlines the step-by-step process to deploy a Streamlit application using Ngrok, allowing external users to access the application without requiring cloud services or setting up a server.

## 1. What is Ngrok?

Ngrok is a tunneling tool that allows you to expose a local web server to the public internet securely. It creates a temporary public URL that can be shared with others.

### Why Use Ngrok?

✔️ No need for a cloud server  
✔️ Simple and quick setup  
✔️ Free version available

## 2. Prerequisites

* Installed Python (>= 3.7)
* Installed Streamlit
* Installed Ngrok

## 3. Installation Steps

### Step 1: Install Streamlit (If Not Installed)

pip install streamlit

### Step 2: Install Ngrok

#### For Linux (Ubuntu/Debian)

wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip

unzip ngrok-stable-linux-amd64.zip

sudo mv ngrok /usr/local/bin/

#### For Windows

* Download Ngrok from [Ngrok Official Website](https://ngrok.com/).
* Extract and place it in a folder, e.g., C:\ngrok.
* Open Command Prompt and navigate to the folder:

cd C:\ngrok

#### For Mac

brew install ngrok/ngrok/ngrok

## 4. Ngrok Authentication

Ngrok requires authentication to use the free tier.

* Sign Up at [Ngrok Sign-Up](https://ngrok.com/signup).
* After signing in, go to "Your Authtoken" in the dashboard.
* Copy the token and run this command:

ngrok authtoken YOUR\_AUTH\_TOKEN

## 5. Running the Streamlit App

Navigate to your Streamlit project folder in the terminal. Run your application on port 8501 (default Streamlit port):

streamlit run app.py --server.port 8501

You should see output like:

Local URL: http://localhost:8501

## 6. Expose Streamlit App Using Ngrok

Open a new terminal window. Run the following command to create a public URL:

ngrok http 8501

Ngrok will display output like:

Forwarding https://abcd-1234.ngrok-free.app -> http://localhost:8501

Copy the generated URL (e.g., https://abcd-1234.ngrok-free.app) and share it with others.

## 7. Keeping the Application Online

⚠️ Important: Ngrok must stay running in the terminal for the public URL to remain active. If you close the terminal, the connection will stop.

## 8. (Optional) Use a Custom Subdomain (Paid Feature)

To get a fixed public URL (instead of random ones), use:

ngrok http -subdomain=myapp 8501

This will give you a permanent link like:

https://myapp.ngrok-free.app

## 9. Pros & Cons of Using Ngrok

| Pros | Cons |
| --- | --- |
| ✅ Free to use for temporary exposure | ❌ Free URLs expire when Ngrok is restarted |
| ✅ No need to buy a domain or cloud hosting | ❌ Limited session duration (6 hours per session for free users) |
| ✅ Simple and quick deployment | ❌ Requires manual restart after disconnection |
| ✅ No port forwarding or router configuration needed | ❌ Limited bandwidth in free tier |

## 10. Alternatives to Ngrok for Free Hosting

If you need permanent hosting, consider:

| Alternative | Best For |
| --- | --- |
| Streamlit Cloud | Hosting public Streamlit apps for free |
| Railway.app | Deploying small applications for free |
| Render.com | Free web services with auto-deployment |
| Vercel / Netlify | Best for frontend applications |

## 11. Conclusion

Using Ngrok, we can quickly deploy a Streamlit application without needing a server or cloud hosting. However, for a long-term solution, using Streamlit Cloud or a free VPS is recommended.

✅ Best for: Quick sharing, testing, and demos  
❌ Not ideal for: Permanent hosting