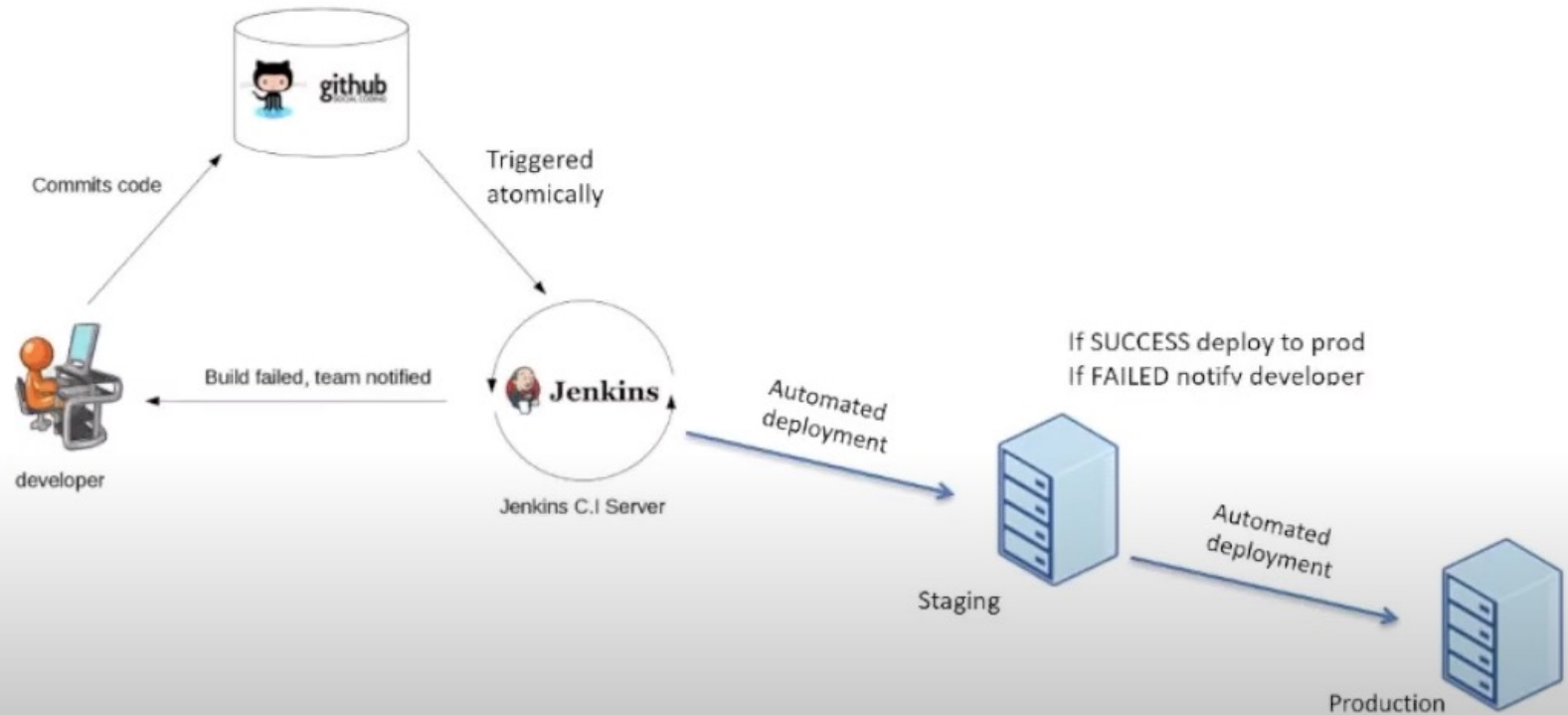
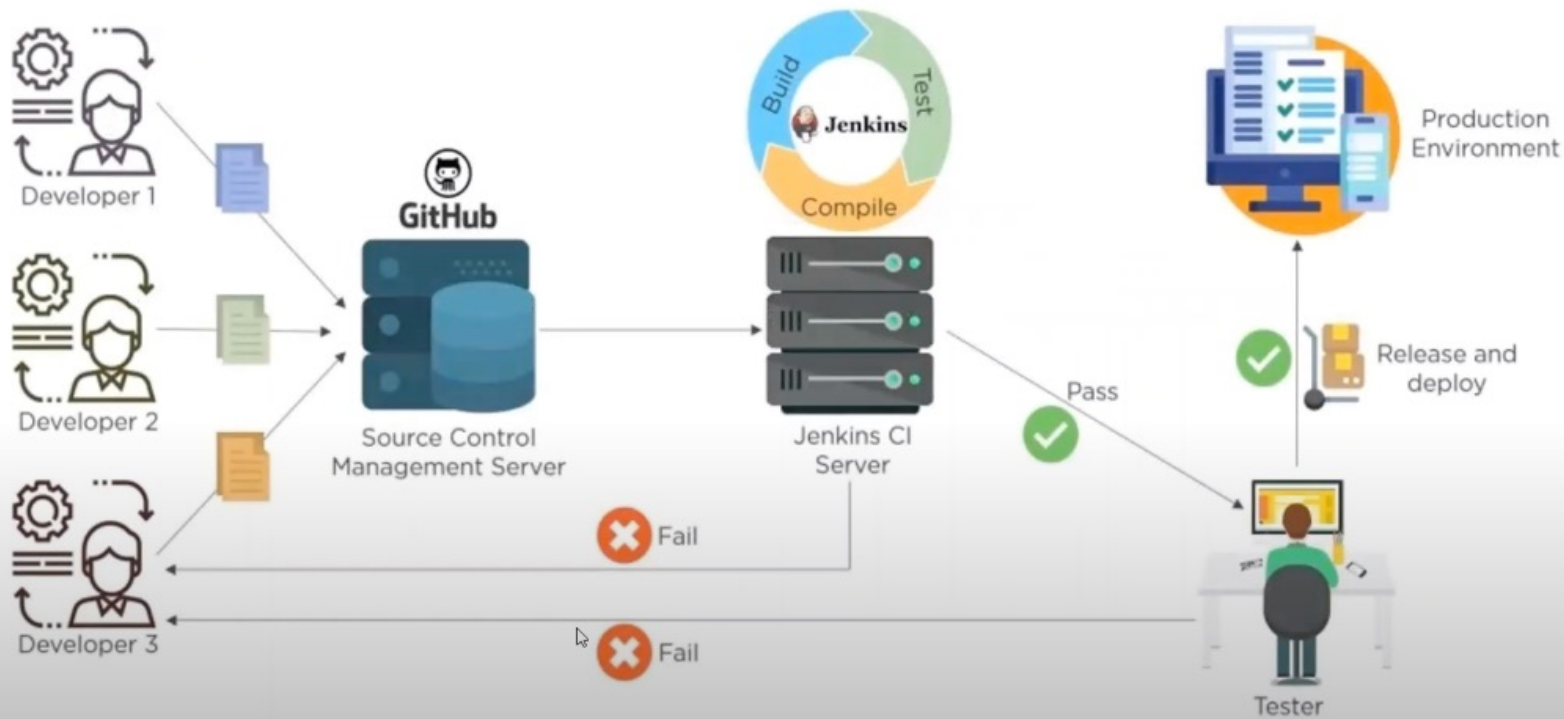


Week 7 : SOFTWARE DEVELOPMENT TOOLS AND ENVIRONMENTS

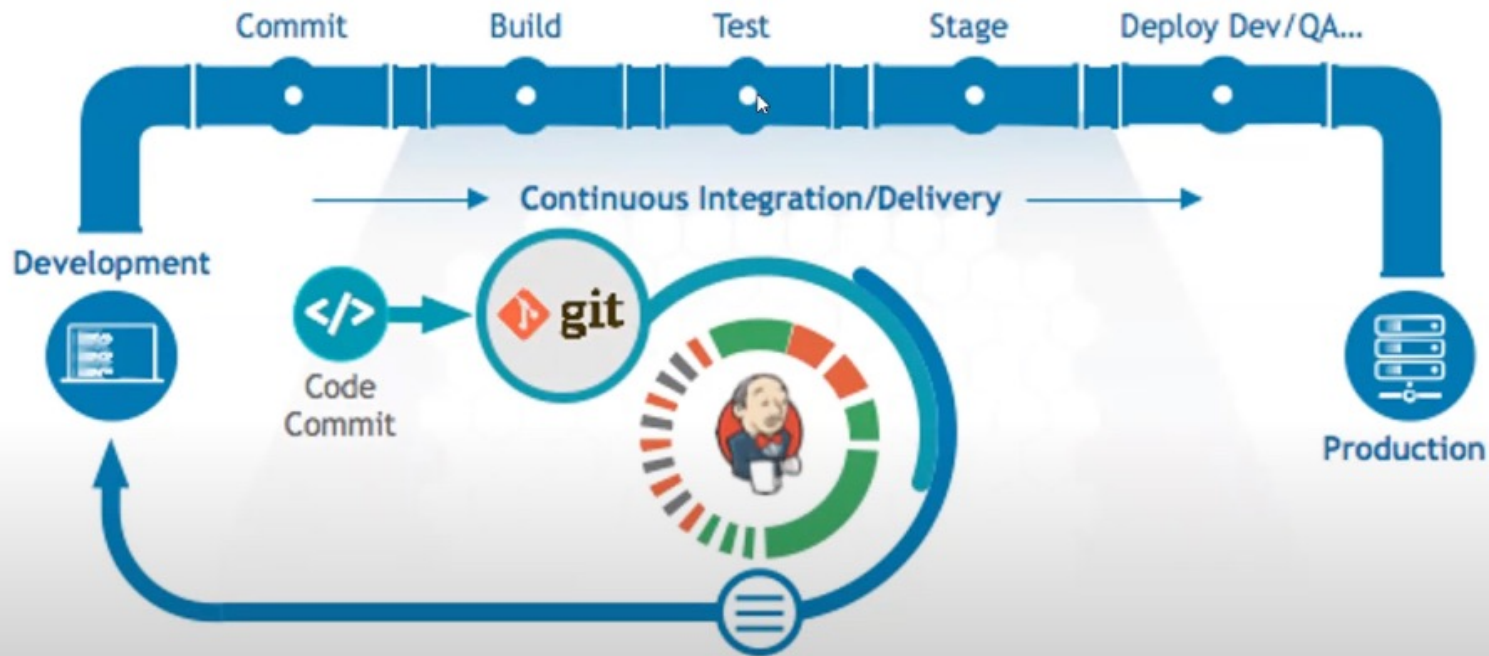


<https://www.youtube.com/watch?v=Ozna0MS5D1E>

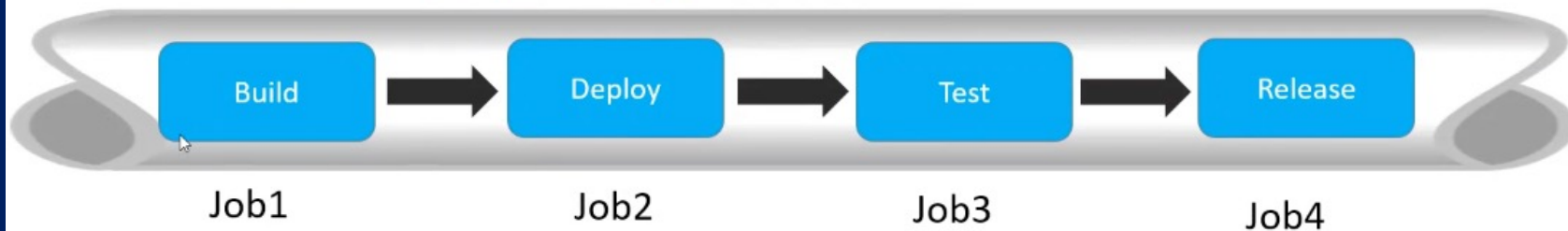
CI & CD



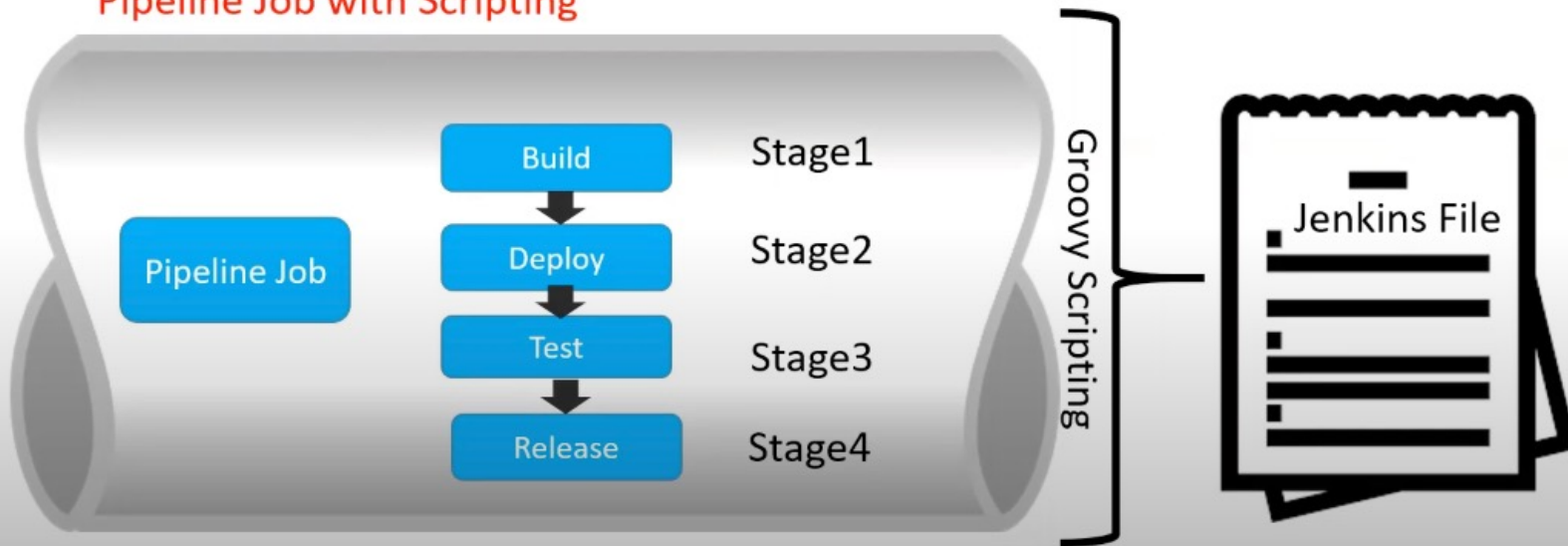
Jenkins Pipeline



Build And Delivery Pipeline Plugins



Pipeline Job with Scripting



Pipeline concepts

- **Pipeline**

- A Pipeline is a user-defined model of a CD pipeline. A Pipeline's code defines your entire build process, which typically includes stages for building an application, testing it and then delivering it.
- Also, *a **pipeline block is a key part of Declarative Pipeline syntax.***

- **Node**

- A node is a machine which is part of the Jenkins environment and is capable of executing a Pipeline.
- Also, *a **node block is a key part of Scripted Pipeline syntax.***

- **Stage**

- A stage block defines a conceptually distinct subset of tasks performed through the entire Pipeline (e.g. "Build", "Test" and "Deploy" stages), which is used by many plugins to visualize or present Jenkins Pipeline status/progress.

- **Step**

- A single task. Fundamentally, a step tells Jenkins what to do at a particular point in time (or "step" in the process). For example, to execute the shell command make use the sh step: sh 'make'. When a plugin extends the Pipeline DSL, that typically means the plugin has implemented a new step.

Scripted Pipeline

Jenkinsfile (Scripted Pipeline)

```
node { ❶
    stage('Build') { ❷
        // ❸
    }
    stage('Test') { ❹
        // ❺
    }
    stage('Deploy') { ❻
        // ❼
    }
}
```

- ❶ Execute this Pipeline or any of its stages, on any available agent.
- ❷ Defines the "Build" stage. `stage` blocks are optional in Scripted Pipeline syntax. However, implementing `stage` blocks in a Scripted Pipeline provides clearer visualization of each stage's subset of tasks/steps in the Jenkins UI.
- ❸ Perform some steps related to the "Build" stage.
- ❹ Defines the "Test" stage.
- ❺ Perform some steps related to the "Test" stage.
- ❻ Defines the "Deploy" stage.
- ❼ Perform some steps related to the "Deploy" stage.

Jenkins Installation on Ubuntu

Installation Steps

1. **Update System Packages** Update the list of available packages and their versions.

```
sudo apt-get update
```



2. **Installation of Java**

```
sudo apt update
sudo apt install fontconfig openjdk-17-jre
java -version
openjdk version "17.0.8" 2023-07-18
OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)
OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)
```



3. **Long Term Support release**

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```



4. **Start Jenkins**

```
sudo systemctl enable jenkins
```



```
sudo systemctl start jenkins
```



```
sudo systemctl status jenkins
```



How to Change Port for Jenkins?



<http://localhost:8080>



<http://localhost:5000>

HTTP Port (default: 8080) – This is the main port used by Jenkins to serve web pages on your machine. By default, this port is set to 8080 but can be changed based on your requirements.

If Jenkins fails to start because a port is in use, run systemctl edit jenkins and add the following:

1. Override the Jenkins Service Configuration

```
sudo systemctl edit jenkins
```



2. Add or Modify the ExecStart Command

```
[Service]  
Environment="JENKINS_PORT=5000"
```



3. Reload the Systemd Daemon

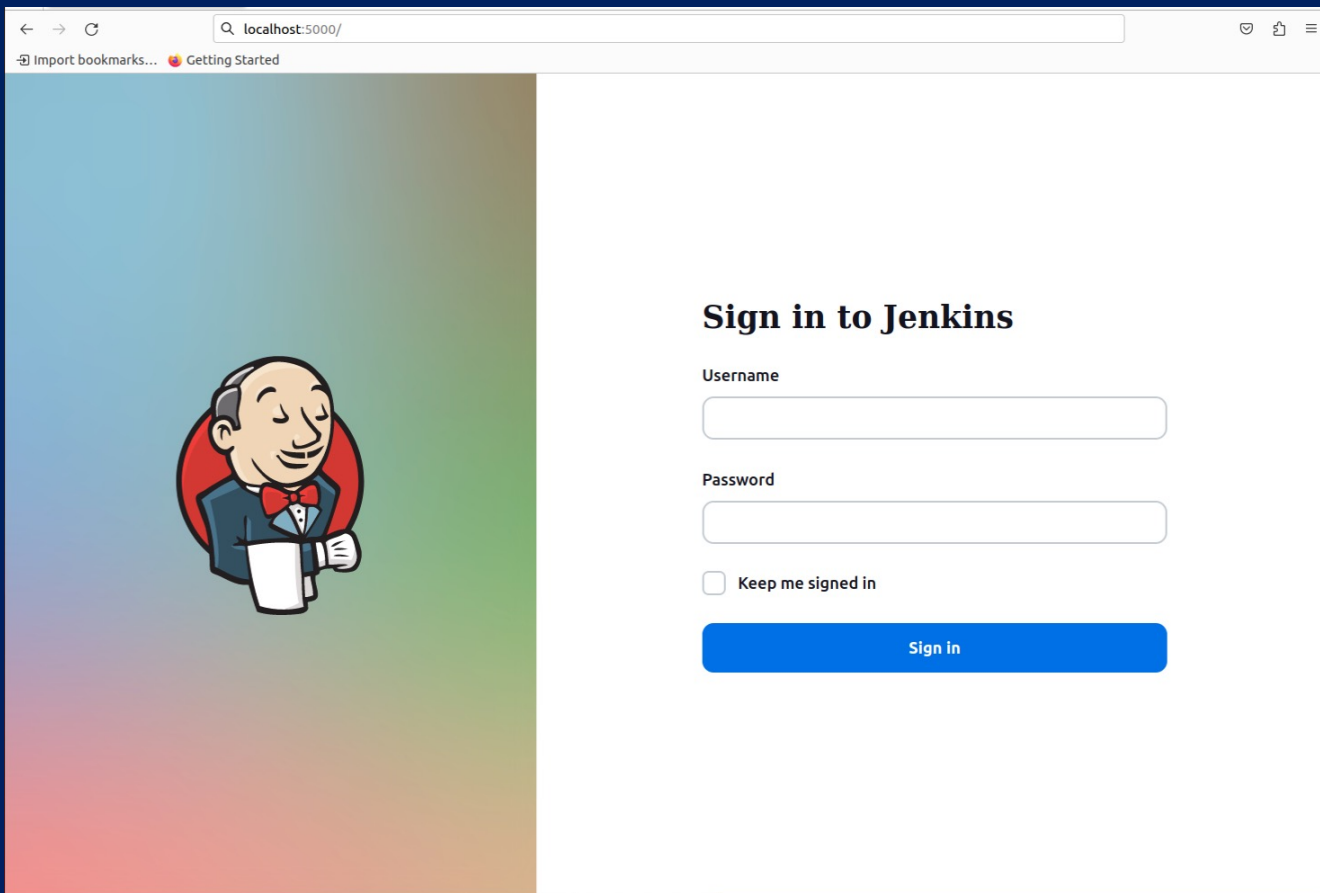
```
systemctl daemon-reload
```



4. Restart the Jenkins service to apply the changes

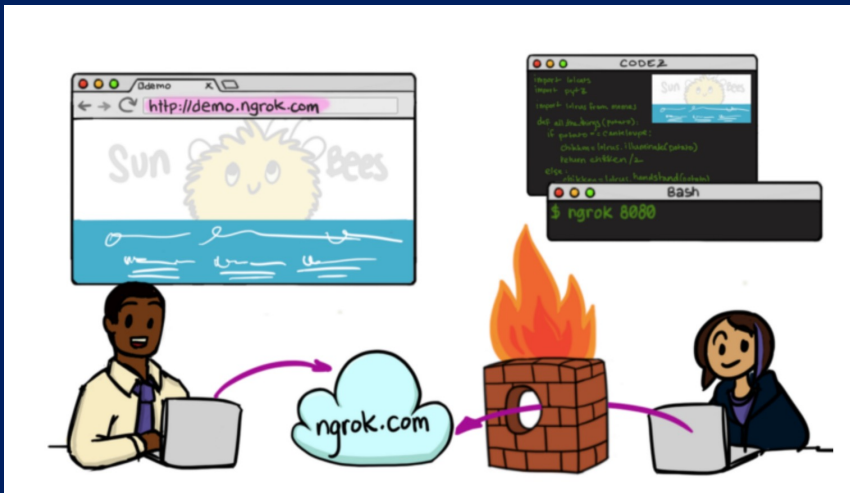
```
sudo service jenkins restart
```





How to forward my local port to public using ngrok

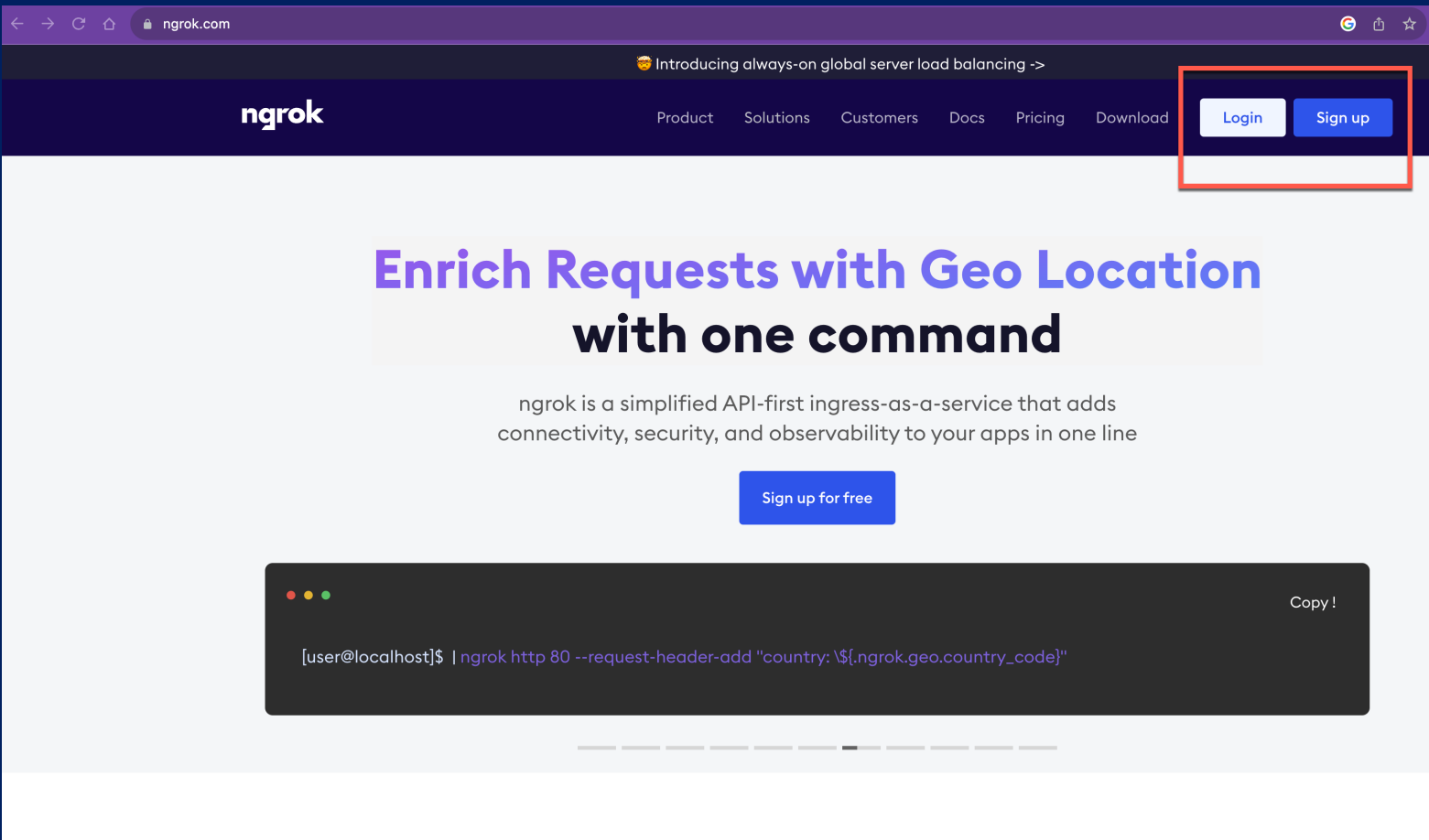
What is ngrok?



ngrok exposes local servers behind NATs and firewalls to the public internet over secure tunnels.

Advantages

- Demoing web sites without deploying
- Building webhook consumers on your dev machine
- Testing mobile apps connected to your locally running backend
- Stable addresses for your connected devices that are deployed in the field
- Running personal cloud services from your home



- 🔔 Getting Started ^
- 🔧 Setup & Installation
- 🔑 Your Authtoken**
- 🌐 Cloud Edge v
- 🔗 Tunnels v
- ⚡ Events
- 🔗 API
- 🛡 Security v
- 👤 Users v
- 💰 Billing
- ⚙ Settings v

Your Authtoken

This is your personal Authtoken. Use this to authenticate the ngrok agent that you downloaded.

To setup ngrok on Ubuntu 22.04, there are two primary methods you can follow. [↗](#)

1. Create an Ngrok Account:

- If you don't have an ngrok account, create one on the [official ngrok website](#).

2. Download ngrok:

- You can download ngrok directly from the Setup & Installation page on the ngrok dashboard or use the following command in the terminal to download the binary file:

```
wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
```



3. Decompress & Install Ngrok Tunnel:

- Extract the downloaded file with the following command:

```
tar zxvf ngrok-v3-stable-linux-amd64.tgz
```



4. Connect Ubuntu to the ngrok account:

- Run the following command to add the auth token to the default Ngrok.yml configuration file (replace the token with your own):

```
./ngrok config add-authtoken <YourAuthToken>
```



5. Usage:

- To create a tunnel to your local server or port forwarding using ngrok, use the following command (replace `8080` with your desired port):

```
./ngrok http 8080
```



- This will provide you with a URL accessible on any device having internet connectivity. The URL of your local Web Interface will be like "<http://127.0.0.1:8080>," and the URL provided by ngrok to access this web server worldwide will be something like "<https://in.ngrok.io>" .

ngrok

Introducing Always-On Global Server Load Balancer: <https://ngrok.com/r/gslb>

Session Status

online

Account 1234567890@gmail.com (Plan: Free)

Version

3.3.5

Region

Asia Pacific (ap)

Latency

39ms

Web Interface

<http://127.0.0.1:4040>

Forwarding

<https://8bc7-171-96-174-205.ngrok-free.app> -> <http://localhost:8080>

Connections

tll	opn	rt1	rt5	p50	p90
2	0	0.01	0.00	30.64	30.92

HTTP Requests

GET /static/e6b141b3/favicon.ico	200 OK
GET /login	200 OK
GET /static/e6b141b3/images/svg/logo.svg	200 OK
GET /static/e6b141b3/jsbundles/simple-page.css	200 OK
GET /	403 Forbidden

← → ↻ 🏠 8bc7-171-96-174-205.ngrok-free.app/login?from=%2F



Sign in to Jenkins

Username

Password

☐ Keep me signed in

Sign in