# References and Useful Links

1. pip install requests
2. https://dashboard.ngrok.com/get-started/setup/windows - ngrok is API gateway to make our application public from localhost.

   Eg: In this project we made private Jenkins - Localhost to public Public - Jenkins which can be accessible from anywhere including github.

***Steps to Set Up ngrok:***

1. **Install ngrok**
   ```
   choco install ngrok     # on Windows (if Chocolatey installed)
   ```
2. **Run ngrok on your Jenkins port** (e.g., 8081)
   ```
   ngrok http 8081
   ```
3. **Copy the public forwarding URL**, like:
   `https://5bff7bfda732.ngrok-free.app/`
4. **Use this as your Payload URL** in GitHub:
   `https://5bff7bfda732.ngrok-free.app/github-webhook/`

✅ Now GitHub can reach your Jenkins!

3. https://www.docker.com/products/docker-desktop/ - To download docker desktop for windows
4. To create a **Docker Hub** account, Go to https://hub.docker.com/signup and register for account creation and use these credentials for login.

### 🎯 After Creating the Account we can push images to Docker Hub with:

- `docker login`
- `docker tag your-image-name your-dockerhub-username/image-name`
- `docker push your-dockerhub-username/image-name`

5. https://chocolatey.org/ - Choco is package manager. Using it we can install many softwares

   ```
   choco install kubernetes-cli
   choco install ngrok
   ```

### *Install Chocolatey (if not already installed)*

Open PowerShell **as Administrator** and run:

```
powershell
CopyEdit
Set-ExecutionPolicy Bypass -Scope Process -Force; `
[System.Net.ServicePointManager]::SecurityProtocol = `
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; `
iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

## 6. ☸ Step-by-Step: Install Kubernetes Dashboard

### ☑ 1. Deploy the Dashboard

Run this command to deploy the official Kubernetes dashboard:
`kubectl apply -f`
https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml

This installs: Dashboard UI Required RBAC roles and service accounts.

### ☑ 2. Create an Admin User (optional but recommended)

Create a YAML file named `dashboard-admin.yaml`:

```
apiVersion: v1
kind: ServiceAccount
metadata:
```

```
  name: admin-user
  namespace: kubernetes-dashboard
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard

kubectl apply -f dashboard-admin.yaml
```

## ☑ 3. Get the Login Token

Run:

```
kubectl -n kubernetes-dashboard create token admin-user
```

Copy the generated token — you'll use it to log in.

## ☑ 4. Start Dashboard Proxy

Run:

```
kubectl proxy
```

Then open this URL in your browser:
http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/

## ☑ 5. Login

Choose **"Token"** authentication.
Paste the token from step 3.

Click **Sign In**.

You now have a full Kubernetes UI running locally, where you can: Monitor deployments, pods, logs, Create/manage workloads, View cluster resources.