🎯 Goal

Build a pipeline using Jenkins that:

1️⃣ Pulls Node.js code from GitHub

2️⃣ Builds Docker image

3️⃣ Pushes image to Docker Hub

4️⃣ Deploys to Kubernetes (Minikube)

5️⃣ Enables Horizontal Pod Autoscaling (HPA)

Structure:

🧱 ARCHITECTURE

scss — Copy code

```
GitHub (NodeJS App)
        ↓
Jenkins Pipeline
        ↓
Docker Build
        ↓
Docker Hub
        ↓
Kubernetes Deployment
        ↓
HPA Auto Scaling
```

🎯 Final Answer

You are actively using:

#	Plugin
1	Git Plugin
2	Docker Pipeline Plugin
3	Pipeline Plugin
4	Credentials Plugin

👉 **Total = 4 Core Plugins**

1. Step:

Install all requisite tools:

- Java

  ```
  sudo apt update
  sudo apt install fontconfig openjdk-21-jre
  java -version
  ```

- Jenkins

  ```
  sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
    https://pkg.jenkins.io/debian-stable/jenkins.io-2026.key
  echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \
    https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
    /etc/apt/sources.list.d/jenkins.list > /dev/null
  sudo apt update
  sudo apt install Jenkins
  ```

- Git

  ```
  sudo apt install git
  ```

- Docker

  ```
  sudo apt remove $(dpkg --get-selections docker.io docker-compose docker-compose-v2
  docker-doc podman-docker containerd runc | cut -f1)

  # Add Docker's official GPG key:
  sudo apt update
  sudo apt install ca-certificates curl
  sudo install -m 0755 -d /etc/apt/keyrings
  sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
  /etc/apt/keyrings/docker.asc
  sudo chmod a+r /etc/apt/keyrings/docker.asc

  # Add the repository to Apt sources:
  sudo tee /etc/apt/sources.list.d/docker.sources <<EOF
  Types: deb
  URIs: https://download.docker.com/linux/ubuntu
  Suites: $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}")
  Components: stable
  Signed-By: /etc/apt/keyrings/docker.asc
  EOF

  sudo apt update

  sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
  compose-plugin
  ```

- Kubernetes

  ```
  sudo apt-get update
  # apt-transport-https may be a dummy package; if so, you can skip that package
  ```

sudo apt-get install -y apt-transport-https ca-certificates curl gnupg
# If the folder `/etc/apt/keyrings` does not exist, it should be created before the curl command, read the note below.
# sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.35/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg # allow unprivileged APT programs to read this keyring
# This overwrites any existing configuration in /etc/apt/sources.list.d/kubernetes.list
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.35/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list # helps tools such as command-not-found to work correctly
sudo apt-get update
sudo apt-get install -y kubectl

- Minikube:
  curl -LO
  https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
  sudo dpkg -i minikube_latest_amd64.deb

2. Step:
   Set Jenkins setup.
   Ip:8080
   Ask for password.
   sudo cat /var/lib/jenkins/secrets/initialAdminPassword
   copy and paste.
   Install suggested plugins.
   Set username and password. Jenkins will be ready.

3. Step:
   Update visudo and assign administration privileges to jenkins user and other accesses
   sudo nano /etc/sudoers
   Add the following line at the end of the file
   jenkins ALL=(ALL) NOPASSWD: ALL
   Jenkins will be accessing the Docker for building the application Docker images, so we need to add the Jenkins user to the docker group.
   sudo usermod -aG docker jenkins
   sudo systemctl restart Jenkins
   After adding the line save and quit the file.
   Now we can use Jenkins as root user and for that run the following command -
   sudo su - jenkins

   Start minikube
   minikube start --driver=docker
   minikube start --memory=4096 --cpus=2
   minikube addons enable metrics-server

kubectl get nodes

ensure Jenkins can run kubectl
sudo cp ~/.kube/config /var/lib/jenkins/
sudo chown jenkins:jenkins /var/lib/jenkins/config

ensure Jenkins have docker hub access.
set credentials and install docker pipeline plugin

in Jenkins account create yaml files.
mkdir k8s
cd k8s
nano deployment.yaml
nano service.yaml
nano hpa.yaml

4. Step:
   Create node js application with npx react-create-app demoapp

5. Step:
   FROM node:18-alpine
   WORKDIR /app
   COPY package*.json ./
   RUN npm install
   COPY . .
   EXPOSE 3000
   CMD ["npm", "start"]

6. Step:
   Test your docker build:
   docker build -t node-app:test .
   docker run -p 3000:3000 node-app:test

7. Step:
   push the code in github

8. Step:
   Create Jenkins project selecting pipeline.
   Make Jenkins groovi script and then build