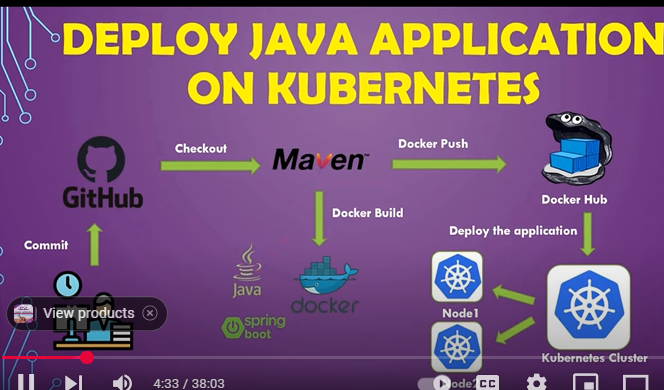
**Deploying Java Applications with Docker and Kubernetes**



1. **Java Application**

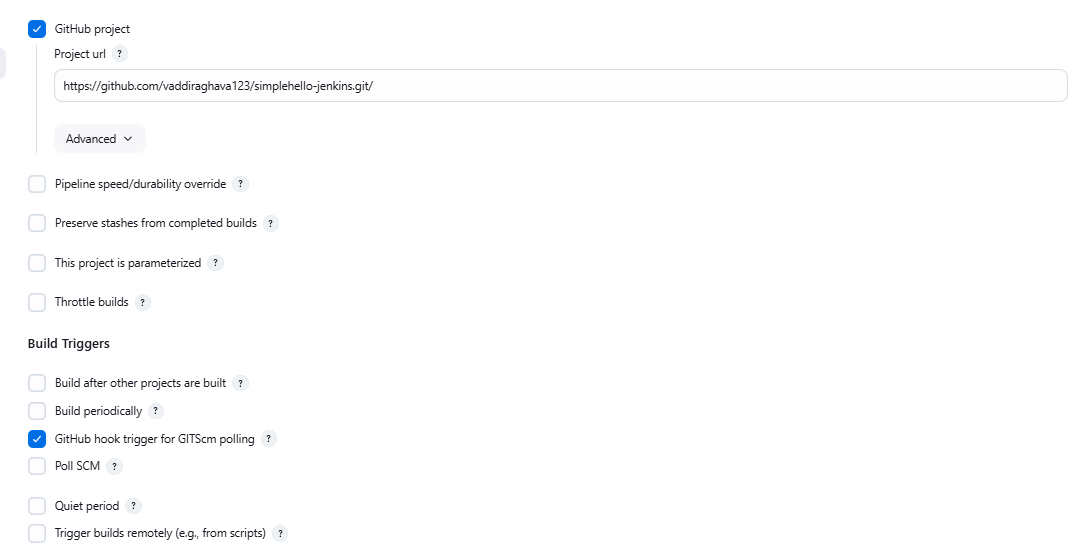
* <https://github.com/vaddiraghava123/simplehello-jenkins>

Simple spring boot java application. Having the rest service “/hello/{name}” and return the message as “

*Hello {name} :Welcome to CI/CD using Jenkins , Docker , K8*

1. ***Note***: Push the repository in always ‘root’ directory like “vaddiraghava123/Repositoryname”. Otherwise, Jenkins is not recognized while pull the code from git hub
2. **Using Jenkins**
   1. Create a ‘New Item’ [helloworld-springautomation] which is left side under ‘Dashboard’ - For Pipeline script
   2. Create a ‘New Item’ [helloworld-springautomation-jenkins] which is left side under ‘Dashboard’ - For Jenkinsfile script

**Pull the GitHub repository**



1. **Two Types of process for processing the CI/CD** 
   1. Through Jenkins file

Jenkinfile

<https://github.com/vaddiraghava123/simplehello-jenkins/blob/main/Jenkinsfile>

* 1. Through pipeline script which is descriptive
     1. Same as Jenkins file
     2. Here we have to take the help of pipeline script for getting configuration of secret password, k8 configuration



**Creation of Docker image**

* Using the Dockerfile, define the steps of image generation

<https://github.com/vaddiraghava123/simplehello-jenkins/blob/main/Dockerfile>

* Note – use bat for windows, sh for linux operating system

**Push the image Docker to hub**

1. Login - Use the pipeline syntax for secret password variables

Here I used directly command

*bat docker login -u vaddiraghava123@gmail.com -p Raghava2025\**

1. Push the image into docker hub

*bat 'docker push vaddiraghava123/spring-hello-integration:latest'*

**Deploy to K8**

Here firstly need to install ‘**Kubernetes deployment plugin’** in Jenkins or use the file and install in Jenkins

https://github.com/vaddiraghava123/simplehello-jenkins/blob/main/ kubernetes-cd.hpi

1. **Pod and service creation**

*bat 'Kubectl run spring-hello-integration --image=vaddiraghava123/spring-hello-integration:latest –port=8081'*

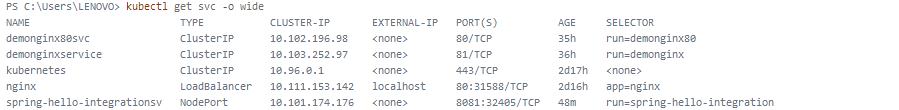
*bat 'kubectl expose pod spring-hello-integration --name=spring-hello-integrationsv --port=8081'*

*Here not used pipeline syntax and not used the Kubernetes plugin which is config the K8(C:\Users\LENOVO\.kube\.config\) copy and paste the file contents. Also, used deployment file of*

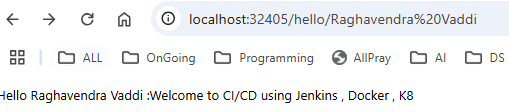
[*https://github.com/vaddiraghava123/simplehello-jenkins/blob/main/deployementservice.yaml*](https://github.com/vaddiraghava123/simplehello-jenkins/blob/main/deployementservice.yaml)

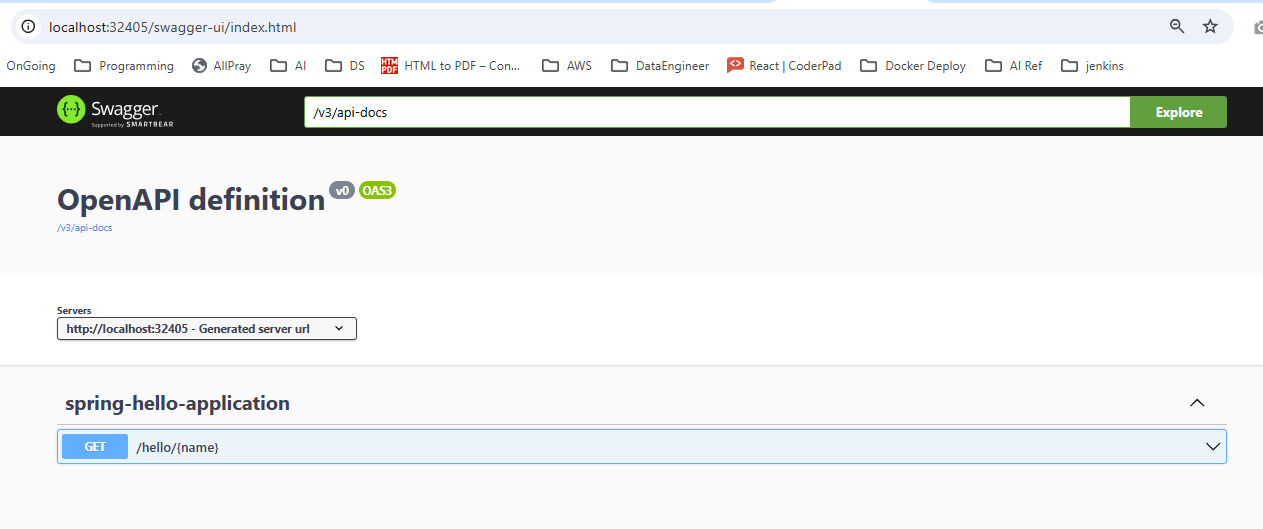
kubernetesDeploy (configs: 'deploymentservice.yaml',kubeconfigId: 'k8sconfigpwd') -- [Not used]

Verify the service before validate



1. **Validate**





**Commands step by step and logs in docker desktop, docker hub, K8**

1. Used ***Pipeline syntax***

checkout scmGit(branches: [[name: '\*/main']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/vaddiraghava123/simplehello-jenkins.git']])

bat "mvn clean install"

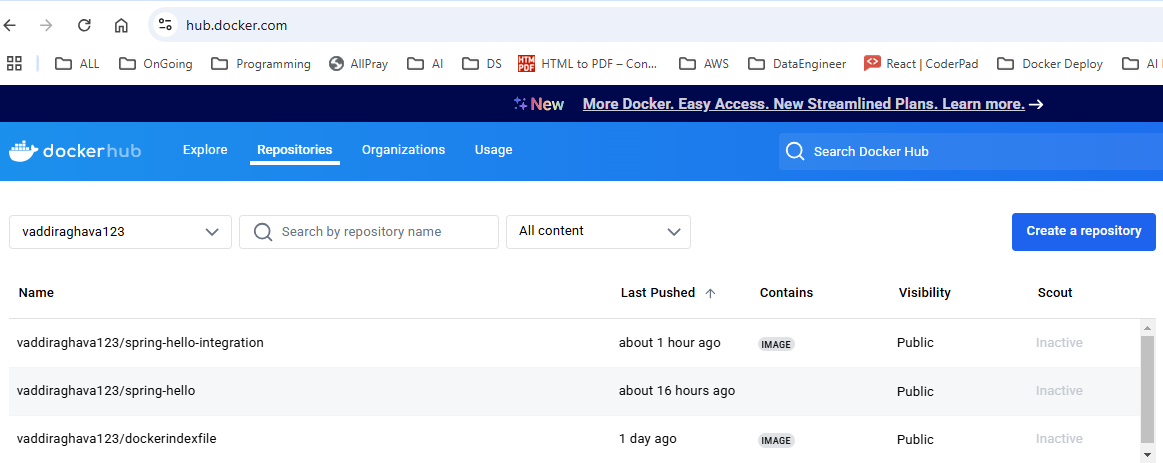
1. bat "docker build -t vaddiraghava123/spring-hello-integration."
2. bat 'docker login -u vaddiraghava123@gmail.com -p \*\*\*\*\*\*'
3. bat 'docker push vaddiraghava123/spring-hello-integration:latest'
4. // kubernetesDeploy (configs: 'deploymentservice.yaml',kubeconfigId: 'k8sconfigpwd') - Through K8 deployment plugin [ Not used ]
5. bat 'Kubectl run spring-hello-integration --image=vaddiraghava123/spring-hello-integration:latest –port=8081'
6. bat 'kubectl expose pod spring-hello-integration --name=spring-hello-integrationsv --port=8081'

**Verify the image in docker hub**

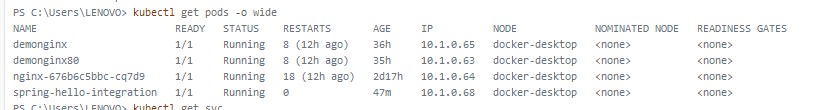
**In service of deploymentservice.yml file, make sure "type": "NodePort"**

Here creates a new repository of image name

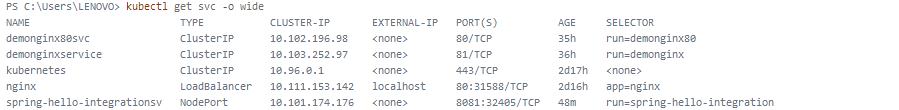
<https://hub.docker.com/>



kubectl get pods -o wide



kubectl get svc -o wide



kubectl logs spring-hello-integration



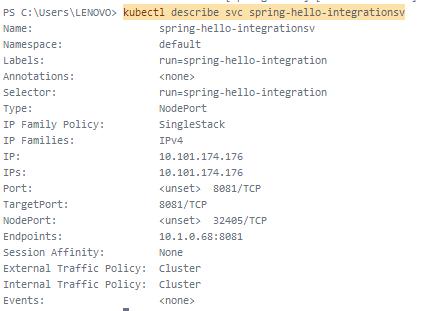
kubectl get endpoints spring-hello-integrationsv



* Edit the service file using kubectl

kubectl edit svc spring-hello-integrationsv

kubectl describe svc spring-hello-integrationsv



**If the NodePort still doesn't work, try this:**

kubectl port-forward svc/spring-hello-integrationsv 8081:8081

**Drop / delete pods and service**

* kubectl delete pods 'spring-hello-integration'
* kubectl delete svc spring-hello-integrationsv