6 Apr 2024

**Real-World Scenario:**

Dr. Shawn runs a pet clinic. He needs to record the visits and other details associated with the pets and their owners visiting his clinic. He has software developed by Bella Solutions, a software company, to manage the same.

Steps to perform this application.

1. Create Simple Spring boot project with starter web starter and one rest method which return welcome message as “Dr. Shawn runs a pet clinic created Akash”.

Create this project using spring initializer

@RestController

Class ClinicController{

@GetMapping(value=”/”)

Public String welcome() {

return “Dr. Shawn runs a pet clinic created Akash”

}

}

Or we can use web starter as well as thymeleaf starter to return index.html page with image and more contents.

1. In application.properties file spring boot run on port number 9090.
2. Create jar file using maven install command.
3. Create Dockerfile

Dockerfile

FROM openjdk:17

COPY ./target/filename.jar .

CMD ["java","-jar","filename.jar"]

Note : please check your jar file name. please check jar file inside a target folder.

1. Now create docker-compose.yml file

version: '3.8'

services:

springboot-container:

build: .

image: spring-boot:v123

container\_name: springboot-container

ports:

- 9090:9090

restart: always

1. Create Jenkinsfile

pipeline {

agent any

tools {

maven "MAVEN"

}

stages {

stage("Build the project"){

steps{

sh "mvn clean"

sh "mvn package"

}

}

stage("run the docker containers"){

steps{

sh "docker-compose down"

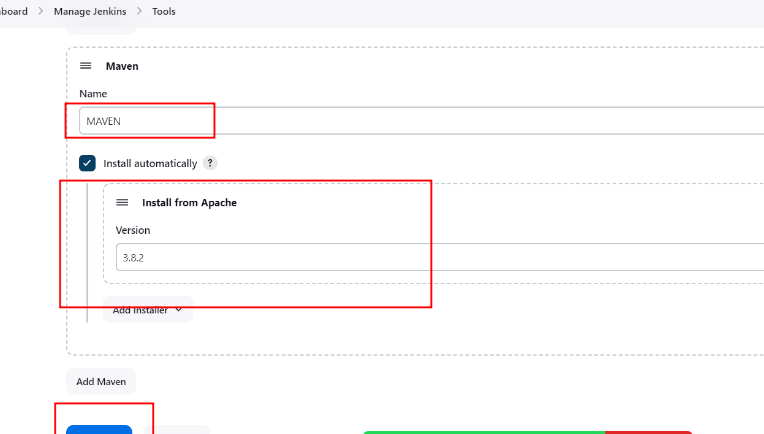
sh "docker-compose up --build -d"

}

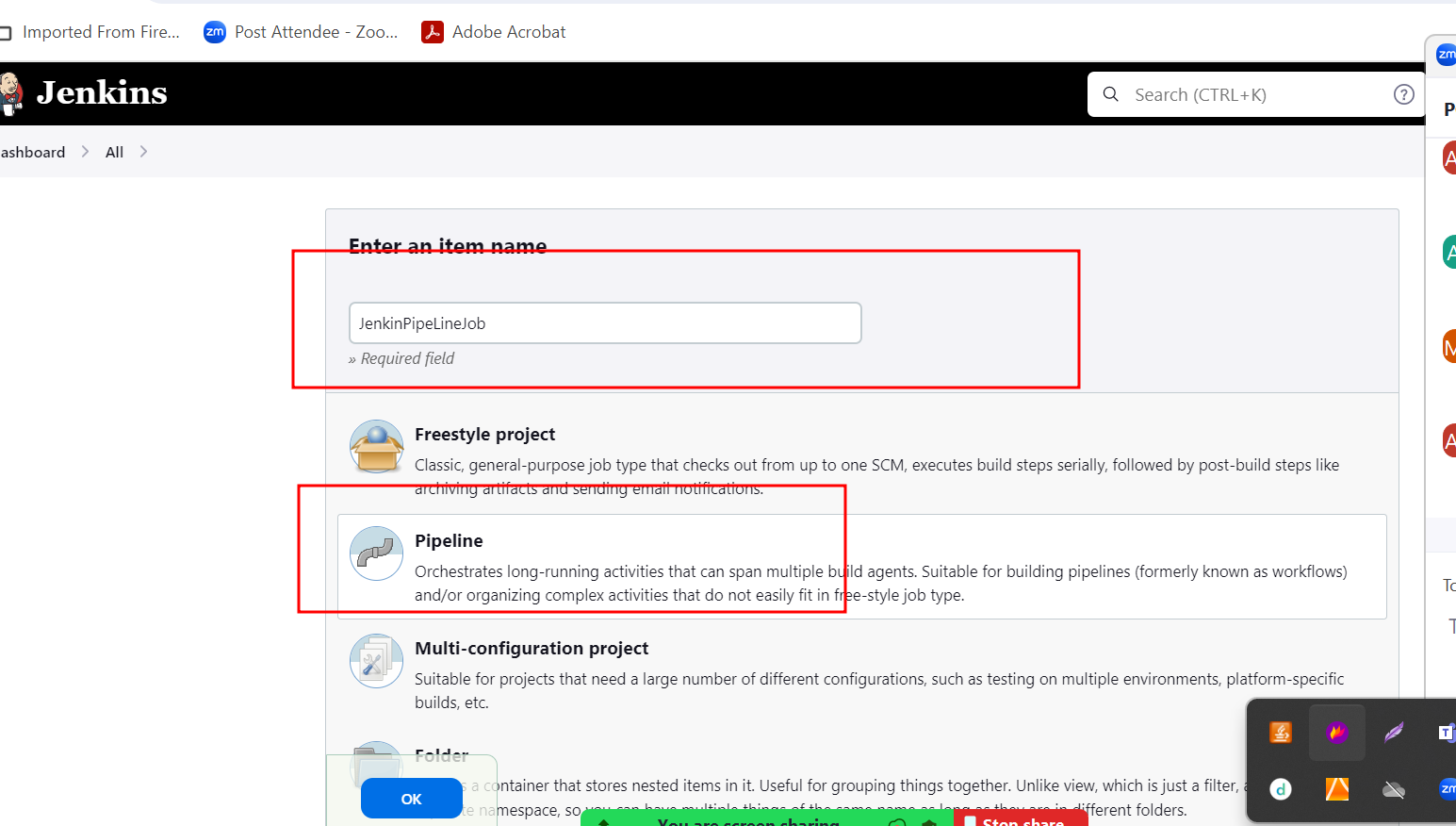
}

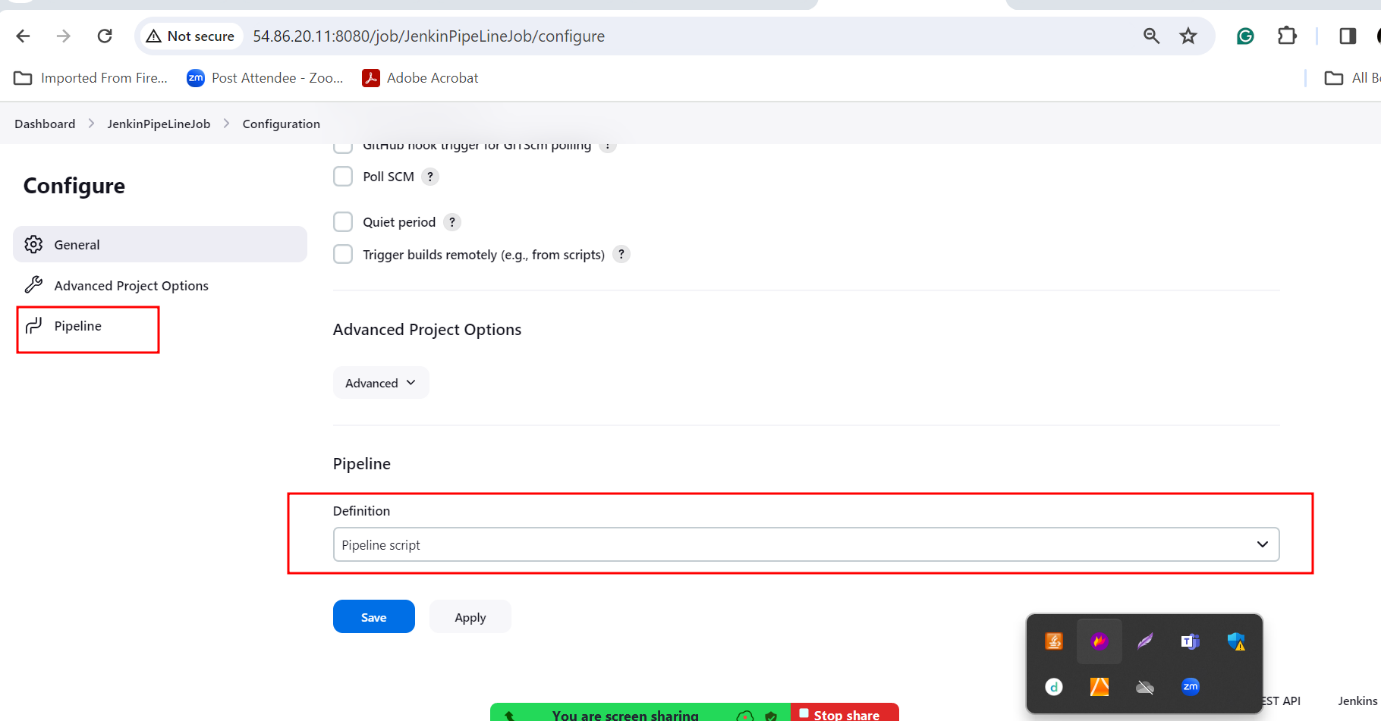
}

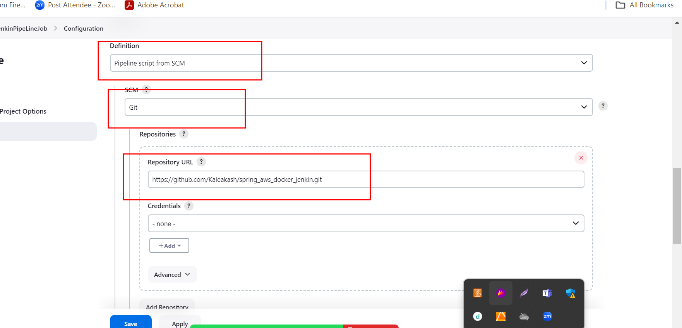
1. Now we need to push the project in our Remote repository. Note : please create new remote repository for course4.
2. Please open command prompt or terminal inside spring boot project.
3. Then write the command as
   1. git init
   2. git add .
   3. git commit -m “initial commit”
   4. please login to git hub account and create new repository.
   5. Create the token and link your local remote repository with github account repository
   6. Push the code.
4. Now login to AWS
5. We create EC2 instance with 2CPU and 4 Gib RAM
6. Open the port number 9090, 8080 (All TCP with range 0 to 65535)
7. Then connect the EC2 instance browser or ssh client
8. Then install git software
9. sudo yum install git -y
10. Then install java
11. sudo yum install java-21 ( install java 21)
12. then we need to install Jenkin
    1. sudo wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat/jenkins.repo>
    2. sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>
    3. sudo yum install jenkins
    4. sudo service jenkins start (java less or 21 version)
    5. sudo cat /var/lib/jenkins/secrets/initialAdminPassword this command to find the password
    6. Run this command on browser <http://IPAddress:8080>
    7. And installed suggested plugin
13. To install docker In EC2 instance terminal
    1. sudo yum install docker
    2. sudo service docker start
    3. sudo docker images
    4. sudo docker run hello-world
14. to install docker compose
    1. sudo curl -L https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
    2. sudo chmod +x /usr/local/bin/docker-compose
    3. docker-compose --version
15. if you want to run docker and docker-compose in jenkin then please execute these commands
    1. sudo usermod -a -G docker jenkins
    2. sudo usermod -a -G docker ec2-user (ec2-user is user name of instance )
    3. sudo chmod 777 /var/run/docker.sock
    4. sudo service jenkins restart
    5. login once again Jenkin dashboard
16. Login to Jenkin dashboard
17. Now we need to configure the maven in Jenkin dashboard
    1. Manage Jenkin
    2. Tools
    3. Maven installation
    4. Add maven
    5. Please provide name (MAVEN)
    6. Version 3.8.X version

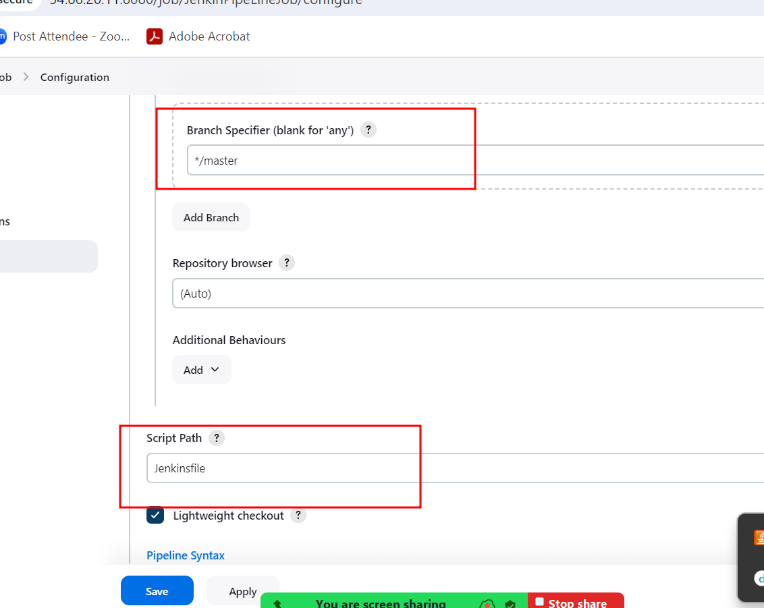


1. Now we create Jenkin pipe line jobs.



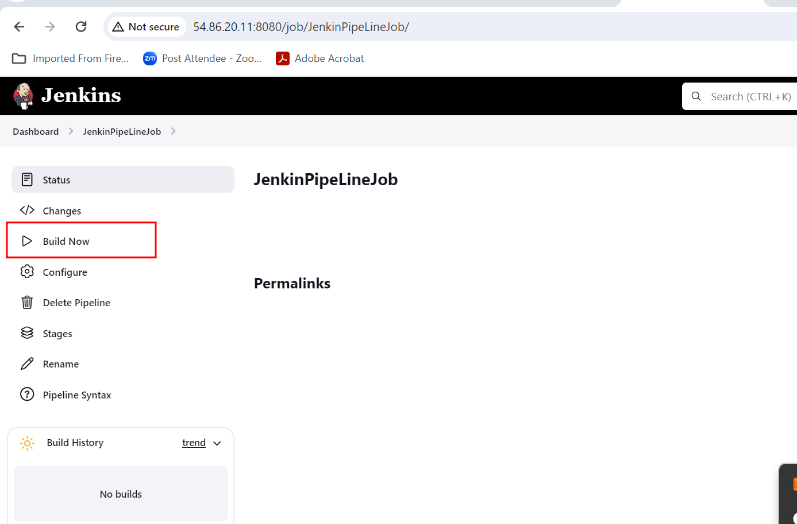






Apply and save

Then build the project and wait



1. If job build successfully
2. <http://publicIpAddress:9090/>
3. If everything go fine you can view your application.

Project submit

1st document

Git hub account URL

Short description about your projects

Which contains Dockefile, docker-compose.yml file and Jenkinsfile

2nd document

Screen short

Output with <http://PublicIPAddress:9090>

3. document

Source code

Rest api

Dockerfile

Docker-compose file

Jenkinsfile