**Setting Up Jenkins Pipeline to Deploy Docker Swarm.**

DESCRIPTION

**Project objective:**

You have to develop an environment for Docker networking.

**Background of the problem statement:**

As you have worked on Docker containers previously, your manager has asked you to perform container scheduling over multiple hosts using Docker CLI and connect multiple hosts with Docker containers.

**You must use the following:**

● Jenkins: To create a pipeline to deploy Docker Swarm  
● Docker Swarm: To implement container networking  
● Git: To connect and push files from the local system to GitHub   
● GitHub: To store the Angular application

**Following requirements should be met:**

● A few of the source code should be tracked on GitHub repositories. You need to document the tracked files that are ignored during the final push to the GitHub repository.  
● Submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository in the document.  
● The step-by-step process involved in completing this task should be documented.

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
| STEP:1 CREATE AWS UBUNTU INSTANCE WITH PORT NO:8080 |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
| 1. Prepare AWS Instance(Ubuntu Server 22.04 LTS (HVM), SSD Volume Type) |
|  |

|  |
| --- |
| 2. Security : Add port no: 80 with Custom TCP Rule |
|  |

|  |
| --- |
| 3. Download .pem Key and connect using Moba x-term |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| open moba x-term |
|  |

|  |
| --- |
| > cd d: //here d: is my drive |
|  |

|  |
| --- |
| > cd phase-5 //phase-5 is a folder inside d: driver where .pen key is available |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| >goto> aws >instance>choose your instance>connect>ssh>copy example key |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
| STEP:2 DOCKER INSTALLATION ON UBUNTU OS |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
| GOTO > GOOGLE> DOCKER INSTALLTION ON UBUNTU |
|  |

|  |
| --- |
| ---------------------------------------------------- |
|  |

|  |
| --- |
| LINK:https://docs.docker.com/engine/install/ubuntu/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| >sudo apt-get update |
|  |

|  |
| --- |
| >sudo apt-get install ca-certificates curl gnupg lsb-release |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| >sudo mkdir -p /etc/apt/keyrings |
|  |

|  |
| --- |
| >curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| >echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| >sudo apt-get update |
|  |

|  |
| --- |
| >sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ------TO VERIFY THE INSTALLTION------ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| > sudo docker -v |
|  |

|  |
| --- |
| output: Docker version 20.10.18, build b40c2f6 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
| STEP:3 INSTALL JDK |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| > sudo apt-get update |
|  |

|  |
| --- |
| > sudo apt install default-jdk -y |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
| STEP:4 INSTALL MAVEN |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| > sudo apt-get update |
|  |

|  |
| --- |
| > sudo apt install maven -y |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
| STEP:5 INSTALL JENKINS |
|  |

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| goto> google> how to download jenkins in ubuntu |
|  |

|  |
| --- |
| link: https://www.jenkins.io/doc/book/installing/linux/ |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| >curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.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 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 1. TO START WITH JENKINS |
|  |

|  |
| --- |
| ------------------------------- |
|  |

|  |
| --- |
| > sudo service jenkins start |
|  |

|  |
| --- |
| > sudo service jenkins status |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| CONNECT: goto> AWS>EC2>Copy Public IP:8080 on browser hit enter |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 2. TO GENERATE SECRET PASSWORD |
|  |

|  |
| --- |
| ---------------------------------- |
|  |

|  |
| --- |
| > sudo cat /var/lib/jenkins/secrets/initialAdminPassword |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| ---to deal with permission denied error---- |
|  |

|  |
| --- |
| > sudo chmod 777 /var/run/docker.sock |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| copy the secret password to jenkins and start installtion |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 3. FOR CONTINUOUS INTEGRATION (CI) |
|  |

|  |
| --- |
| -------------------------------------- |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 1. click on install suggested plugins |
|  |

|  |
| --- |
| 2. provide your credentials |
|  |

|  |
| --- |
| 3. welcome to jenkins |
|  |

|  |
| --- |
| 4. manage plugin>maven integration> click on install without restart |
|  |

|  |
| --- |
| 5. goto>dashboard>create>new job>select freestyle project> give name>click ok |
|  |

|  |
| --- |
| 6. give description |
|  |

|  |
| --- |
| 7. sourcecode management |
|  |

|  |
| --- |
| GIT: |
|  |

|  |
| --- |
| URL: https://github.com/Nikunj-Java/docker\_master.git |
|  |

|  |
| --- |
| BRANCHES TO BUILD: \*/master or \*/main |
|  |

|  |
| --- |
| 8. Build Trigger |
|  |

|  |
| --- |
| POLL SCM: |
|  |

|  |
| --- |
| H/2 \* \* \* \* |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 9. CLICK ON APPLY AND SAVE |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 10. BUILD THE PROJECT |
|  |

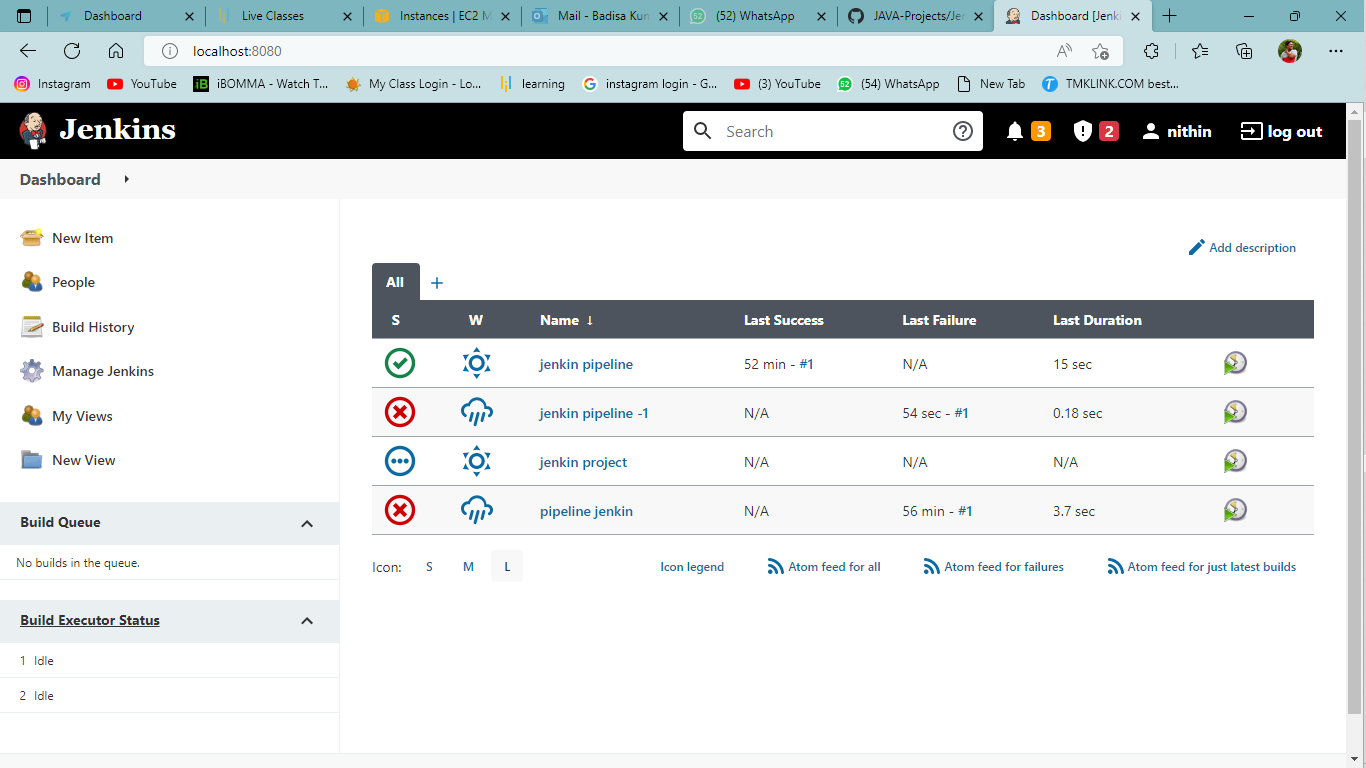
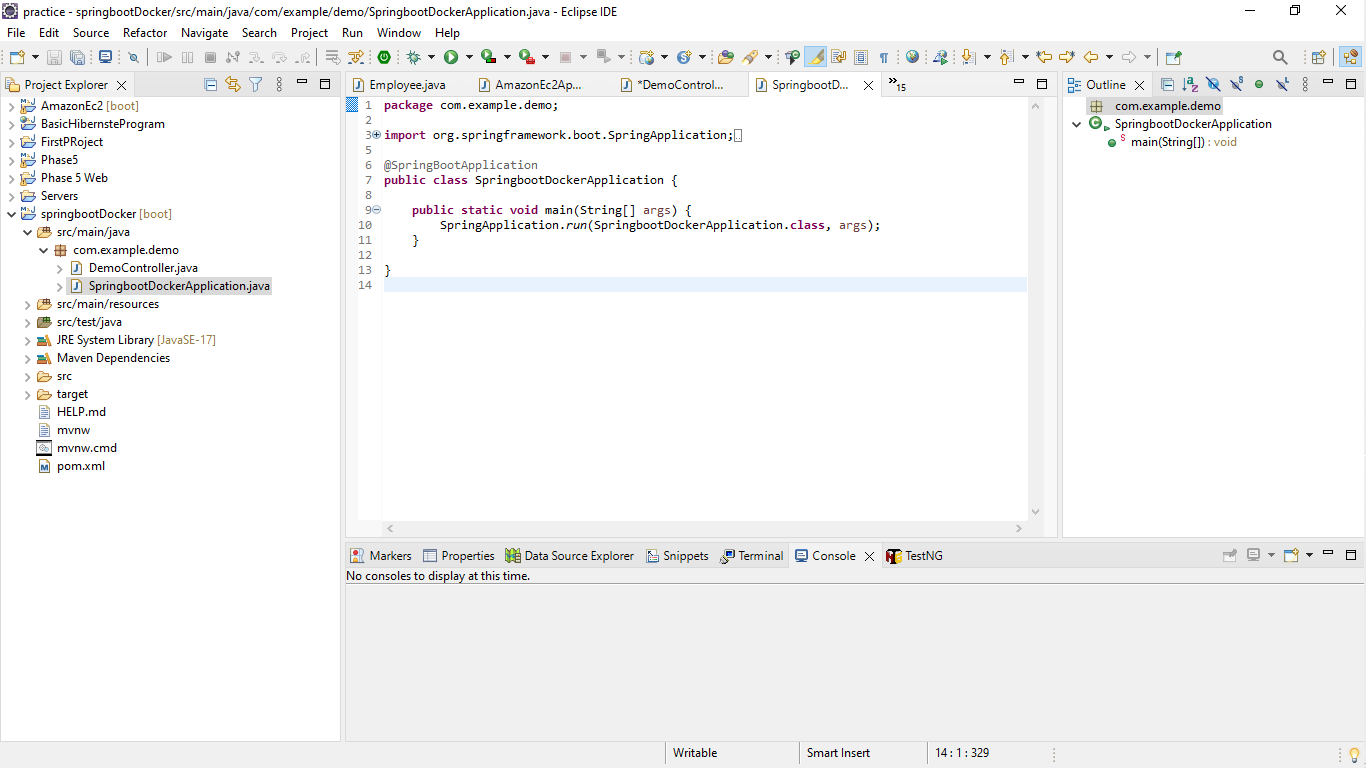
|  |
| --- |
|  |
|  |

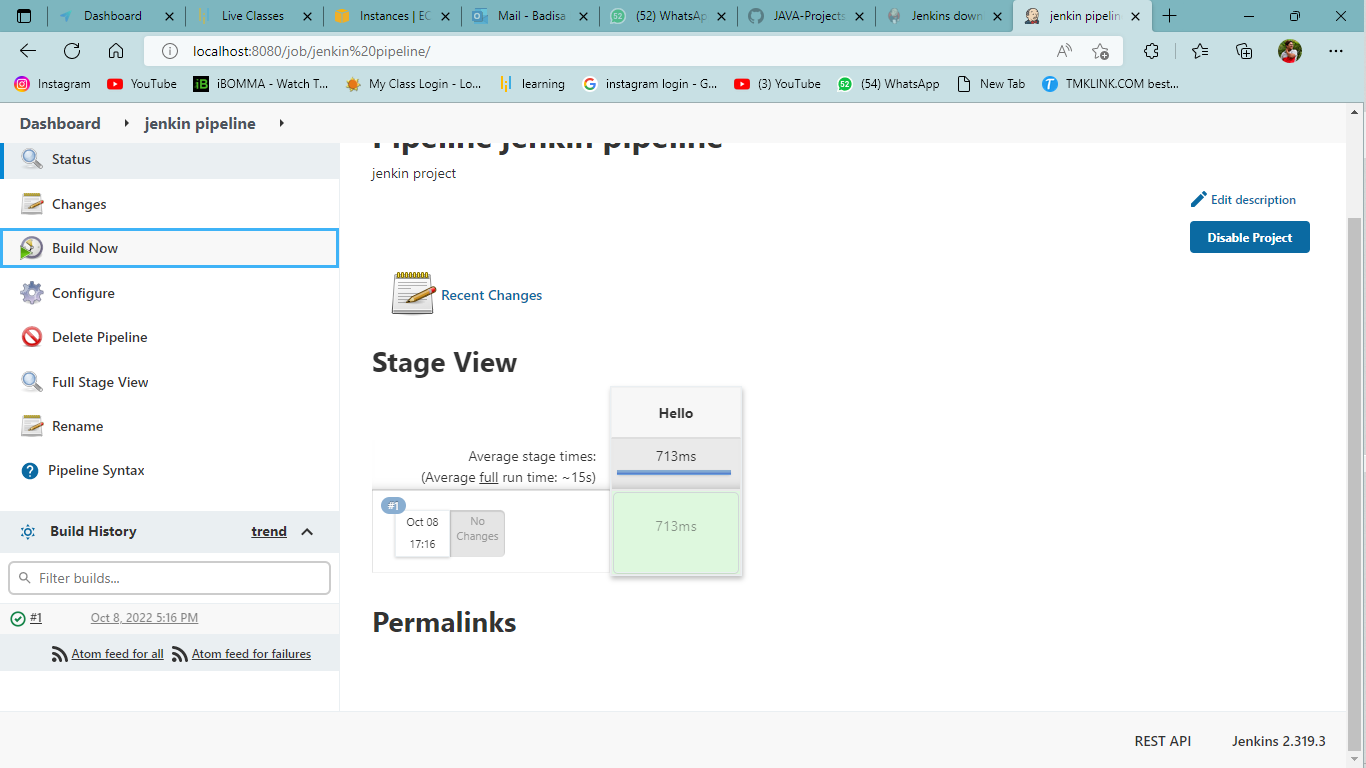
|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |

****

****