Installing Jenkins using Docker containers on linux

Run the following into the Dockerfile on linux machine:

```
FROM jenkins/jenkins:2.414.2-jdk17
USER root
RUN apt-get update && apt-get install -y lsb-release python3-pip
RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc \
  https://download.docker.com/linux/debian/gpg
RUN echo "deb [arch=$(dpkg --print-architecture) \
  signed-by=/usr/share/keyrings/docker-archive-keyring.asc] \
  https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
RUN apt-get update && apt-get install -y docker-ce-cli
USER jenkins
RUN jenkins-plugin-cli --plugins "blueocean:1.25.3 docker-workflow:1.28"
```

Build the image from this:

```
docker build -t myjenkins-app-1 .
```

Create docker network:

```
docker network create jenkins
```

Creating docker container:

```
docker run --name jenkins-app --restart=on-failure --detach \
  --network jenkins --env DOCKER_HOST=tcp://docker:2376 \
  --env DOCKER_CERT_PATH=/certs/client --env DOCKER_TLS_VERIFY=1 \
  --publish 8080:8080 --publish 50000:50000 \
  --volume jenkins-data:/var/jenkins_home \
  --volume jenkins-docker-certs:/certs/client:ro \
  myjenkins-app-1
```

Check the docker container:

```
docker ps
```

login to web browser:

```
127.0.0.1:8080
```

Execute the below cmd to list PWD for jenkins

```
docker exec jenkins-app cat /var/jenkins_home/secrets/initialAdminPassword
```

--- copy the password & paste it into the web browser.

While installation:

```
# install suggested plugins.

# give
- username
```

- password
- name
- save & continue.

DEMO - 1:

On Web browser

- Browse through the UI.
- Create a new/first job
- Without filling GitHub URL.
- In BUILD, select "*execute shell*" & write some commands like:
    - echo "Hello world"
        - Save & build project
        - Check the console logs
- Then edit the shell again with environmental variables
    - echo "Build ID is: ${BUILD_ID}"
    - echo "Build URL is: ${BUILD_URL}"
        - Save & build project
        - Check the console logs
- Then edit the shell again:
    - Create file and verify it on browser & container location as well.
        - echo "this is Jenkins Pipeline" > ReadMe.txt
        - Save & build project
        - Check the console logs

Now, login to docker container

To access the file system in jenkins container within docker

```
docker exec -it jenkins-app bash
```

Execute commands:

```
# pwd
# ls -lrt
# cat ReadMe.txt
```

 # go to
/var/jenkins_home/workspace/my-first-proj-1

DEMO - 2 :- Fetching info from GitHub account

- Create a new job.
- Provide the GitHub URL in the source code management under 'repository URL'.
  https://github.com/jitendrastomar5593/gitdemo
- In Build, select 'execute shell' & run below command.
  Python3 helloworld.py
  - Save & check.
  - Build it.
  - Check console.

==========
Dashboard --> Manage Jenkins

--> Plugins

- Install blueOcean plugins.

--> Manage Nodes & Clouds --> cloud providers

- Docker
- Amazon EC2 (time taking process)
    - Check Install after restart.

Integrating GitHub with Jenkins on AWS EC2 (Ubuntu)

Saturday, March 16, 2024 11:16 PM

1. Create an Ubuntu VM on any cloud (AWS) with port number
   a. 80
   b. 443
   c. 8080 (for Jenkins application)
   d. 22
2. Connect to the EC2 instance using SSH via
   a. Putty
   b. MT Putty
   c. MobaXterm
   d. PowerShell
   e. Windows Terminal
3. Update the ubuntu server and install docker on it.
   a. Set up Docker's apt repository

```
# Add Docker's official GPG key:
sudo apt-get update -y
sudo apt-get install ca-certificates curl -y
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:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update -y
```

   b. Install docker packages

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

   c. Verify that the Docker Engine installation is successfully

```
docker run hello-world
```

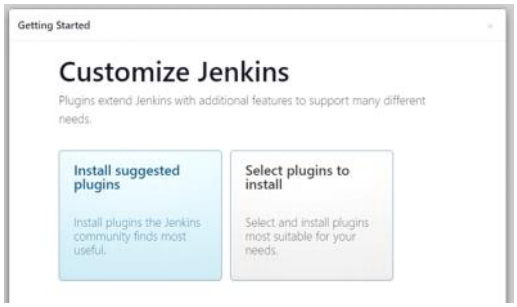
      # systemctl enable docker
      # systemctl start docker
      # usermod -a -G docker <ubuntu>

   d. Pulling Jenkins image from dockerhub.
      # docker pull jenkins/jenkins
   e. Listing images:
      # docker images
   f. Create a new directory:
      # mkdir Jenkins
   g. Running Jenkins container on port 8080
      # docker run -d --name jenkins -p 8080:8080 -v $PWD/jenkins/ jenkins/jenkins
      # docker ps
   h. Login to <EC2-INSTANCE-PUBLIC-IP>:<PORT-NUMBER>
      http://<ec-instance>:<8080>
      Unlock the Jenkins using
      # docker ps
      # docker exec -it <e580724f6f66> cat /var/jenkins_home/secrets/initialAdminPassword

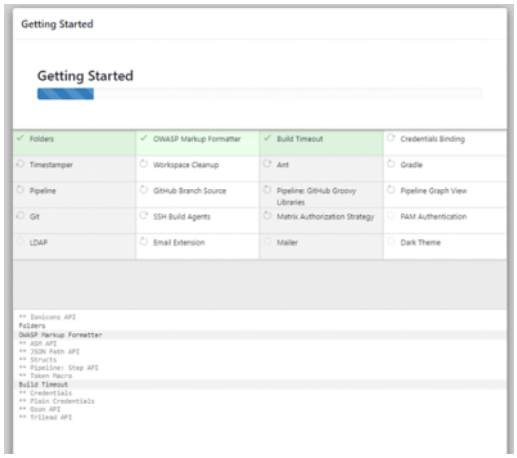
```
root@ip-172-31-41-164:~# docker exec -it e3d072d2f2f4 cat /var/jenkins_home/secrets/initialAdminPassword
d97cfe56c27641788b4935120e644b51
```

Copy this password and paste it to browser page.

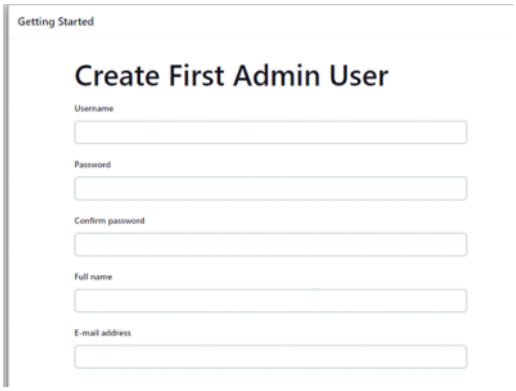
Select **install suggested plugins**:



Wait for it:



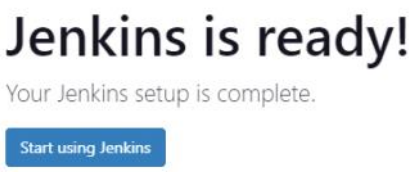
Fill the below details:



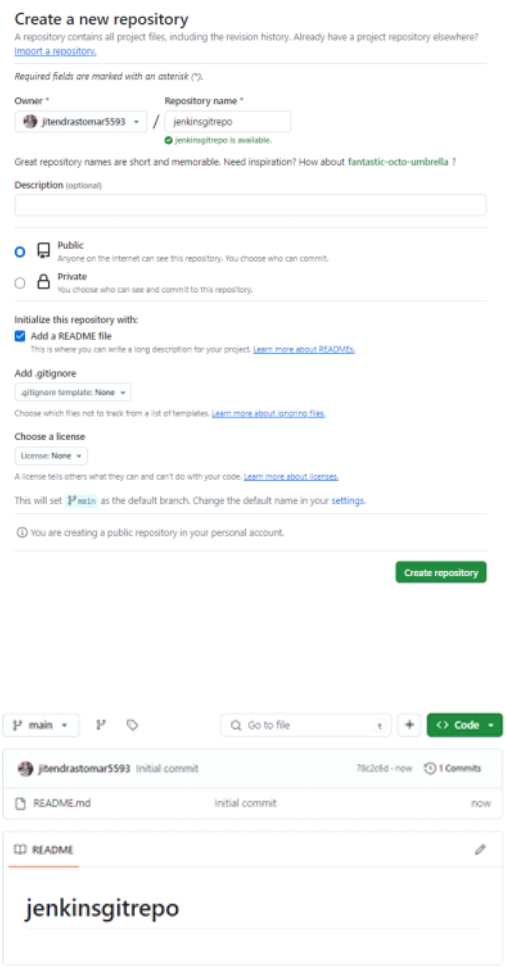
Save & continue.

Check the URL: http://<IP-OF-EC2-INSTANCE>:8080/

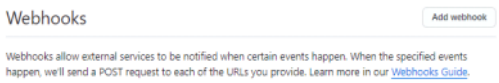
Save & finish.



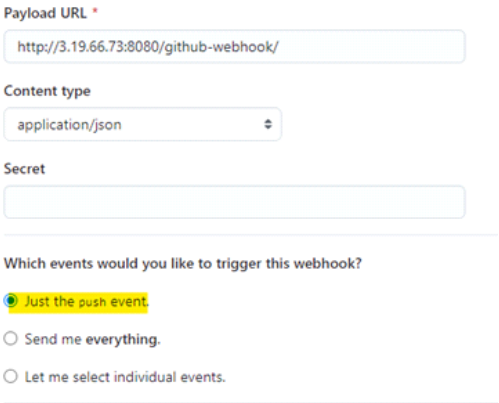
Switch to GitHub & create a new public repository



On GitHub, go to “settings” & then “webhooks”. Click “Add webhook”.



Fill details:



Payload URL → http://<jenkins-svr-public-IP>:<port>/github-webhook/

Content type → application/json

Which events would you like to trigger this webhook → Just the push event

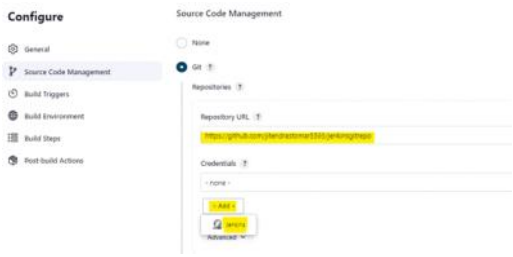
Ensure “Active” is check marked. & click on “Add webhook”. & wait for some time.

Now switch back Jenkins portal, Jenkins dashboard: - Create “new item”.

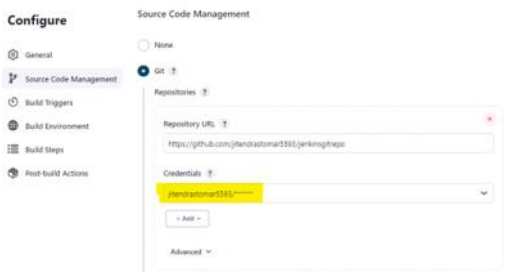
Enter an item name:

Click OK.

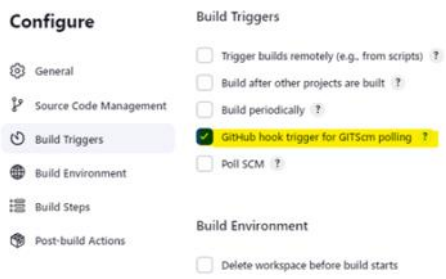
Under “Configure” select “source code management”, select GIT & fill ‘repository URL’.



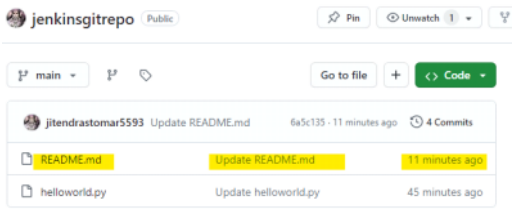
Select “Jenkins” & under credentials, select username & password & fill those.



Click on “Build Triggers”. Select GitHub hook.



Click “Build Steps”, Nothing to select and save. Switch back to GitHub & edit the README.md file & commit it.



Go back to jenkins dashboard page & wait for build history to show up, automatically.

Creating another file & commit on GitHub & switch to Jenkins dashboard and visit build history.

From <https://d.docs.live.net/4b07f4230dd85219/Documents/Jenkins%20documentation%20-%20working%20docx.docx>