Day 23 : Jenkins Freestyle Project

This is #90DaysofDevops challenge under the guidance of Shubham Londhe sir.

Day 23 TASK

check this for task:

https://github.com/LondheShubham153/90DaysOfDevOps/blob/master/2023/day23/tasks.md

What is CI/CD?

- CI or Continuous Integration is the practice of automating the integration of code changes from multiple developers into a single codebase. It is a software development practice where the developers commit their work frequently into the central code repository (Github or Stash). Then there are automated tools that build the newly committed code and do a code review, etc as required upon integration. The key goals of Continuous Integration are to find and address bugs quicker, make the process of integrating code across a team of developers easier, improve software quality and reduce the time it takes to release new feature updates.
- CD or Continuous Delivery is carried out after Continuous Integration to make sure that we can release new changes to our customers quickly in an error-free way. This includes running integration and regression tests in the staging area (similar to the production environment) so that the final release is not broken in production. It ensures to automate the release process so that we have a release-ready product at all times and we can deploy our application at any point in time.

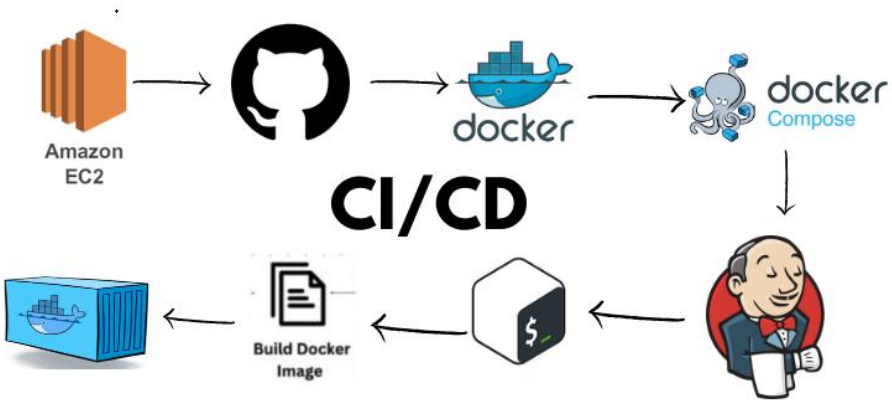
What Is a Build Job?

A Jenkins build job contains the configuration for automating a specific task or step in the application building process. These tasks include gathering dependencies, compiling, archiving, or transforming code, and testing and deploying code in different environments.

Jenkins supports several types of build jobs, such as freestyle projects, pipelines, multi-configuration projects, folders, multibranch pipelines, and organization folders.

What is Freestyle Projects ?

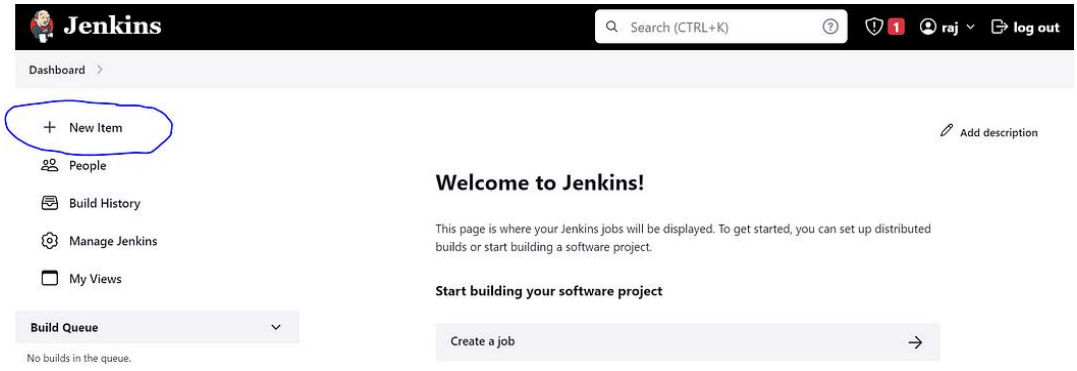
A freestyle project in Jenkins is a type of project that allows you to build, test, and deploy software using a variety of different options and configurations. Here are a few tasks that you could complete when working with a freestyle project in Jenkins:



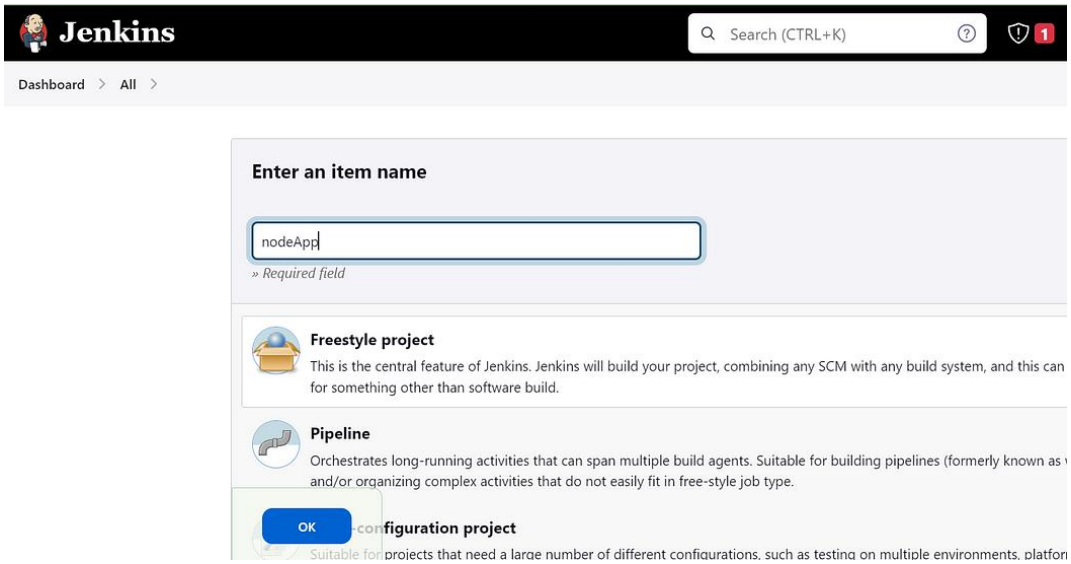
Task-01

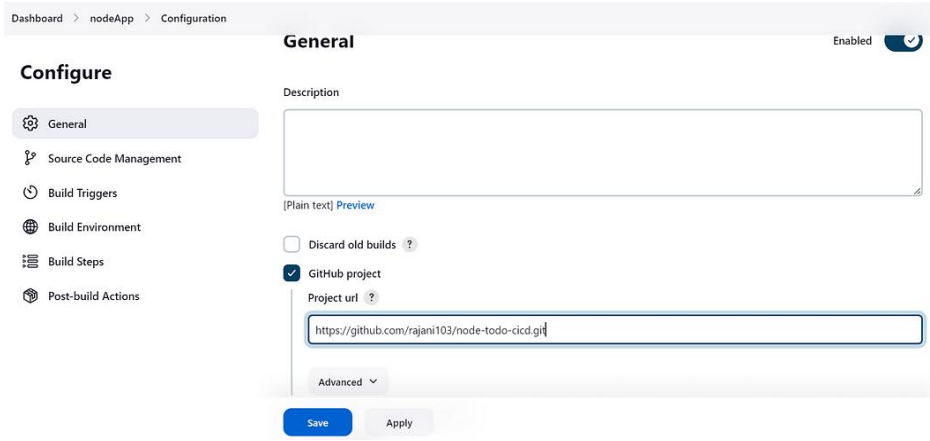
Create a agent for your app.

Create a new Jenkins freestyle project for your app.

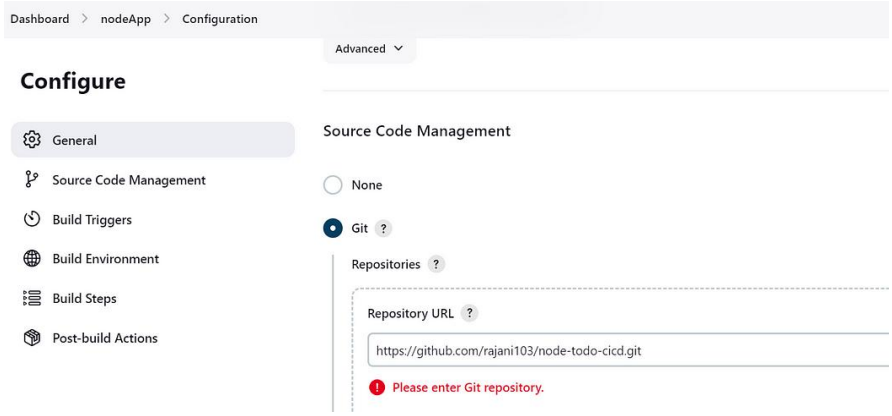
- Log in to Jenkins and open the Dashboard
- Cretae a “New Item”
- 

- Select “Freestyle project” and give name to your project.
-

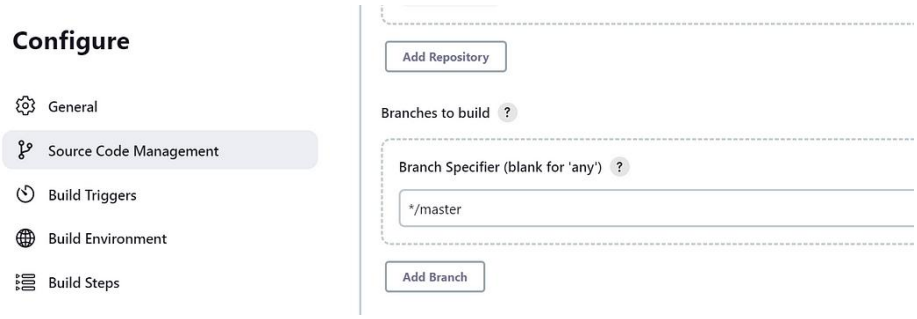


- Click on the “OK” button to create the project.
- In the project configuration, put the details of the project, such as the source code management system, build triggers, and build actions. In GitHub project put your GitHub project repository URL.
- 

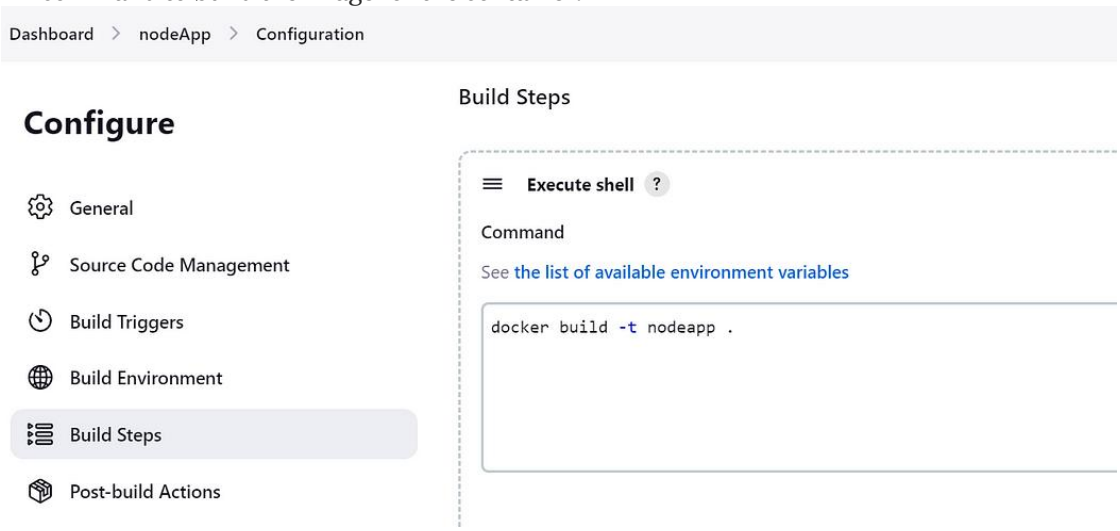
In the “Source Code Management” section,add your repository Link.



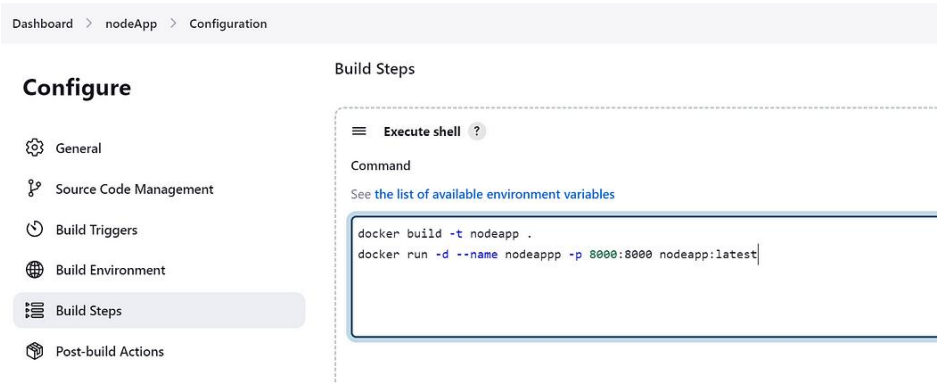
Specify branch where the source code is present.



- In the “Build” section of the project, add a build step to run the “docker build” command to build the image for the container.



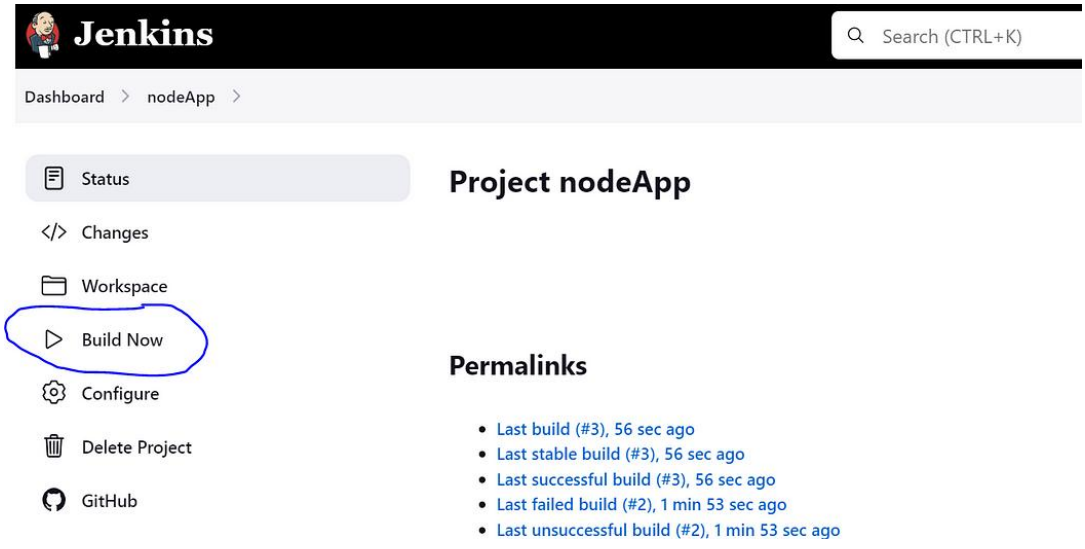
- Add a second step to run the “docker run” command to start a container using the image created in step 3.

- 

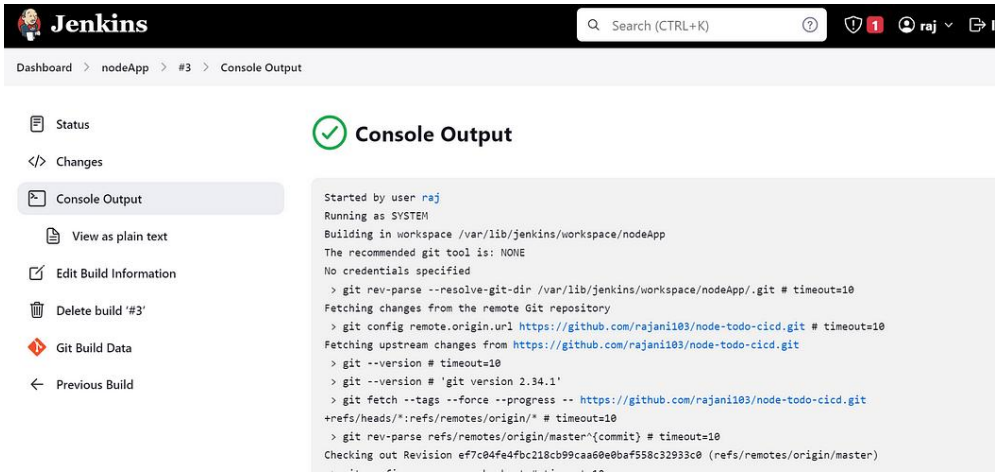
```
docker build -t nodeapp .
docker run -d --name nodeappp -p 8000:8000 nodeapp:latestSave the project configurations by clicking on the save button.
```

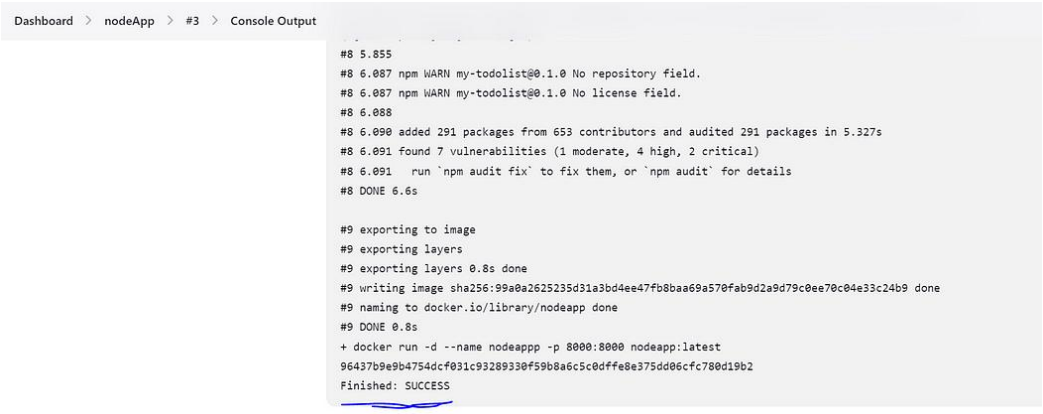
Now you can start the build process of the project by manually clicking on the “build now” tab.

This will start the build process and execute the steps specified in the project configuration (execute shell).



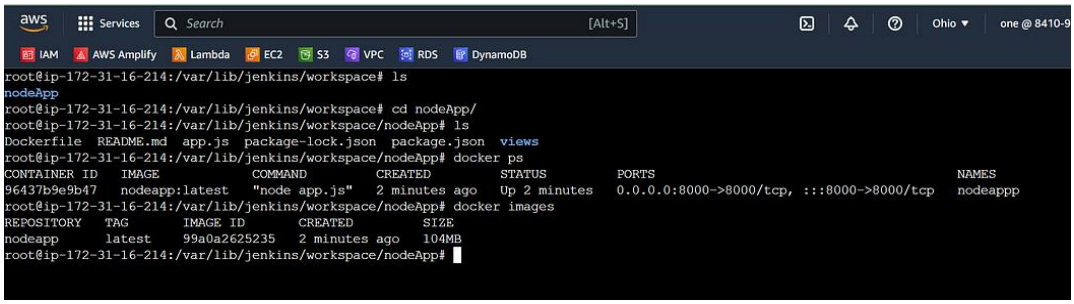
Once the build is successful , you can go to console output nad check the output of the build.





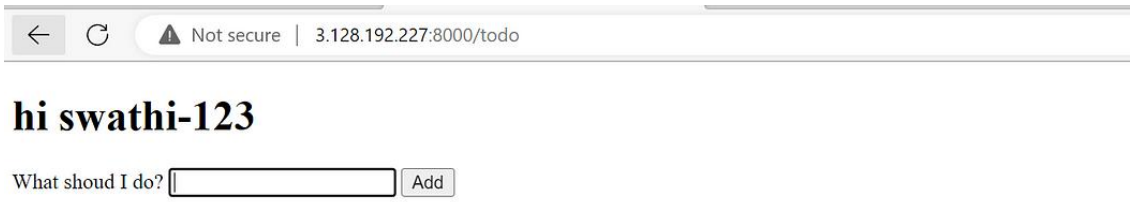
You can also go to the EC2 instance and check that the source code is been cloned on the VM and you can check the same by going to this location :

/var/lib/jenkins/workspace



Also you can see image has been created and a container has been launched using the image that has been built, using the commands in the execute shell.

Try accessing the application using the public IP and the 8000 port on the browser.



Task-02

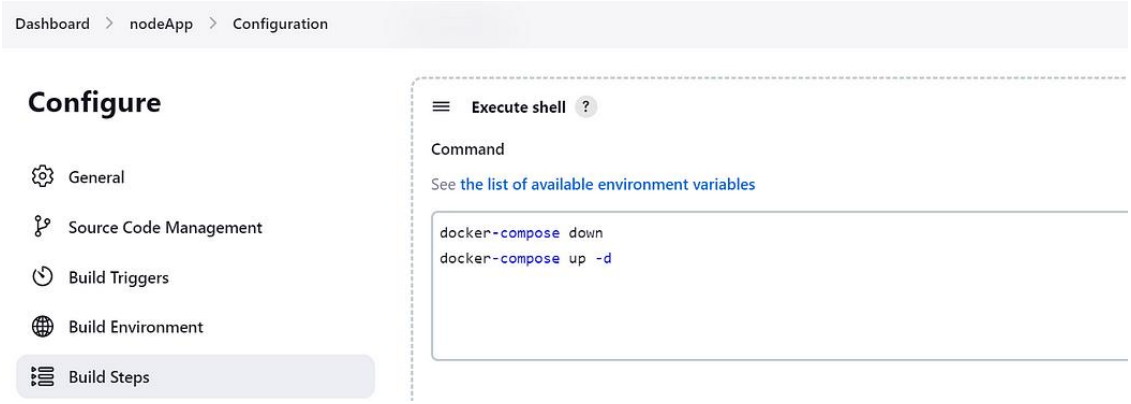
- Create Jenkins project to run “docker-compose up -d” command to start the multiple containers defined in the compose file.

- Create docker-compose.yml file inside your project folder.

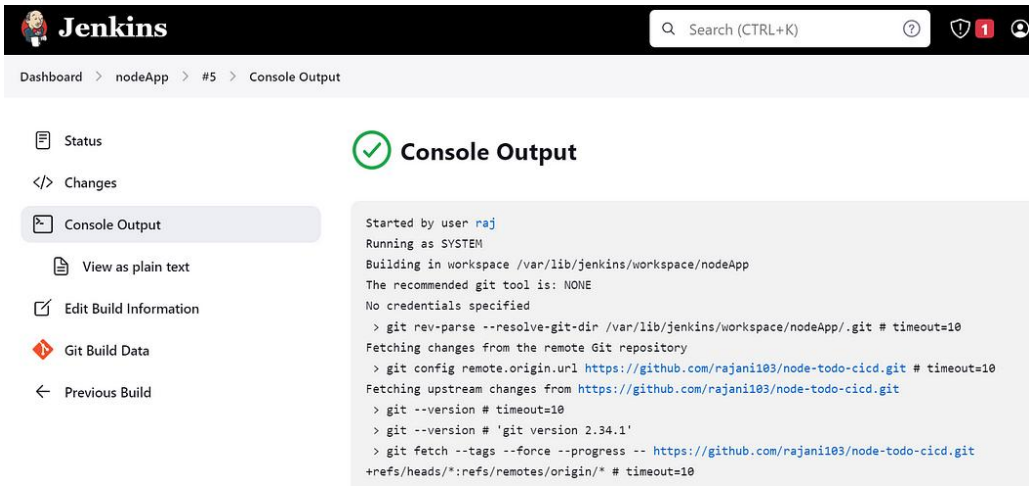
```
root@ip-172-31-16-214:/var/lib/jenkins/workspace/nodeApp# cat docker-compose.yml
version : "3.3"
services:
  web:
    build : .
    ports:
      - "8001:8000"
  db:
    image: mysql
    ports:
      - "3306:3306"
```

- In the “Build” section of the project, add a build step “docker-compose down” command to stop and remove the containers defined in the compose file, then add “docker-compose up -d” command.

- Set up a cleanup step in the Jenkins project to run “docker-compose down” command to stop and remove the containers defined in the compose file.



Save the configurations and build the project.



Dashboard > nodeApp > #5 > Console Output

```
#9 exporting to image
#9 exporting layers
#9 exporting layers 0.8s done
#9 writing image sha256:0ac1b1993c93f813aad593a977078b613a55a653bd4148ae28ff00ba1c8817f3 done
#9 naming to docker.io/library/nodeapp_web done
#9 DONE 0.8s
Image for service web was built because it did not already exist. To rebuild this image you must use `docker-
compose build` or `docker-compose up --build`.
Pulling db (mysql:)...
latest: Pulling from library/mysql
Digest: sha256:e8dc80474af00e0c49d55742815da52073f466059648738e648c9dd262ff51db
Status: Downloaded newer image for mysql:latest
Creating nodeapp_web_1 ...
Creating nodeapp_db_1  ...
Creating nodeapp_web_1 ... done
Creating nodeapp_db_1  ... done
Finished: SUCCESS
```

You can check in the instance that the containers has been created.

```
root@ip-172-31-16-214:/var/lib/jenkins/workspace/nodeApp# docker ps
CONTAINER ID   IMAGE            COMMAND          CREATED              STATUS              PORTS                                          NAMES
ae51826e13ec   nodeapp_web      "node app.js"    About a minute ago   Up About a minute   0.0.0.0:8001->8000/tcp, :::8001->8000/tcp      nodeapp_web_1
96437b9e9b47   nodeapp:latest   "node app.js"    19 minutes ago       Up 19 minutes       0.0.0.0:8000->8000/tcp, :::8000->8000/tcp      nodeappp
root@ip-172-31-16-214:/var/lib/jenkins/workspace/nodeApp#
```

Now you can try accessing the application on browser.

My shaandaar todolist peepp1 | My shaandaar todolist peepp1

Not secure | 3.128.192.227:8001/todo

hi swathi-123

What shoud I do? [ ] Add

Please, feel free to drop any questions in the comments below. I would be happy to answer them.

If this post was helpful, please do follow and click the clap

_Thank you for reading

_Rajani