

[ DevOps-Task-3 ]

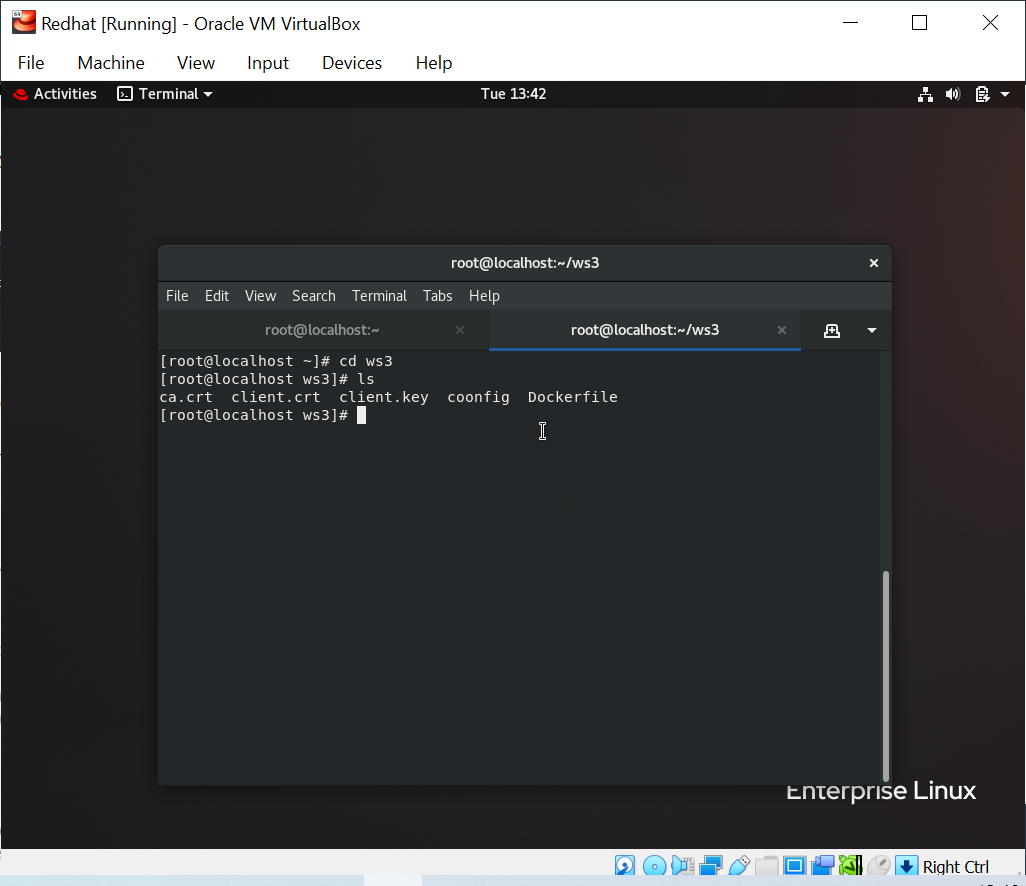
**Management and Deployment of Webserver/App using Jenkins & Docker on top of Kubernetes.**

Perform the second task on top of Kubernetes where we use Kubernetes resources like Pods, Replica Set, Deployment, PVC, and Service.

* Create container image that’s has Jenkins installed using Docker file Or You can use the Jenkins Server on RHEL 8/7
* When we launch this image, it should automatically start the Jenkins service in the container.
* Create a job chain of **job1**, **job2**, **job3**, and **job4** using the build pipeline plugin in Jenkins.
* **Job-1**: Pull the GitHub repo automatically when some developers push the repo to GitHub.
* **Job-2** :  By looking at the code or program file, Jenkins should automatically start the respective language interpreter installed image container to deploy code on top of Kubernetes ( If code is of PHP, then Jenkins should start the container that has PHP already installed).
* Expose your pod so that the testing team could perform the testing on the pod.
* Make the data to remain persistent (If the server collects some data like logs, other user information).
* **Job-3**: Test your app if it is working or not.
* **Job-4**: if the app is not working, then send an email to the developer with error messages and redeploy the application after code is being edited by the developer.

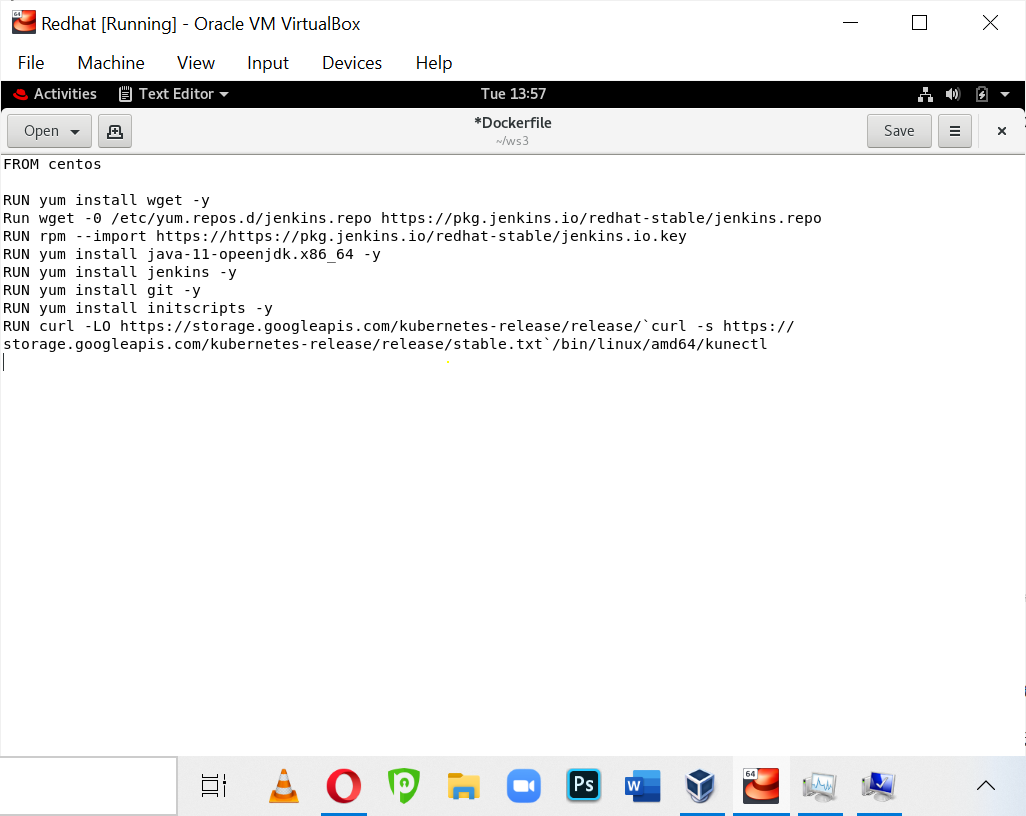
**Get started...**

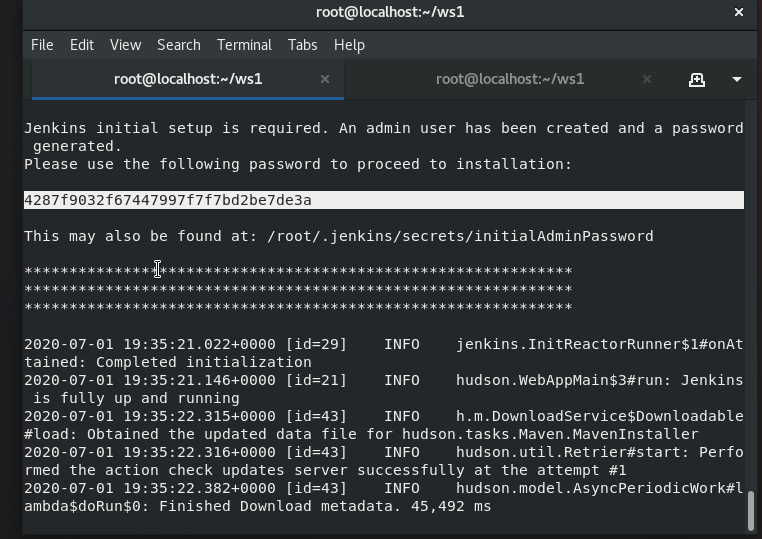
Launch the RHEL8 node, which will be the host for Docker. Create a folder here, for Docker file. This folder must also have ca.crt, **client**.**crt**, **client**.**key**, and config file of **minikube** aka Kubernetes running on Windows. Copy these files using WinSCP. These files are needed to Kubernetes in Docker containers.



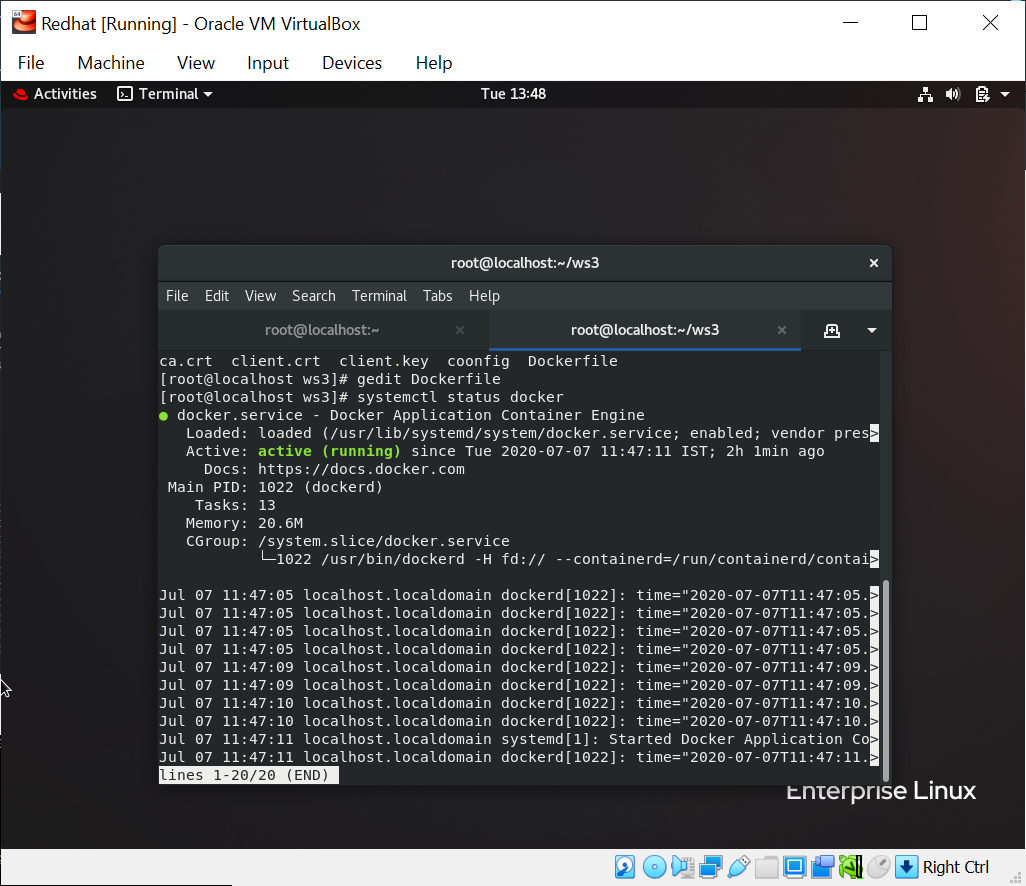
This command is only when Docker file must be in same directory.

Code for the docker image which will setup Jenkins server and also Kubernetes in our container.

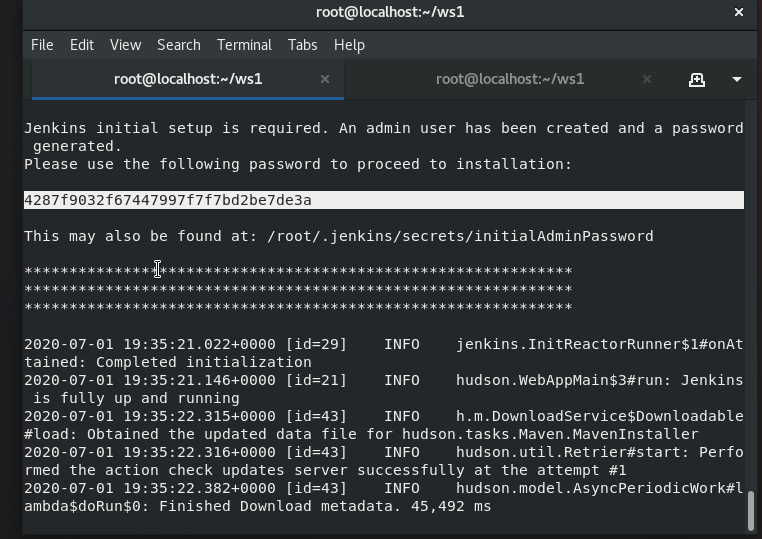


enabled.

Check docker status. To make sure that container it’s running and working fine.

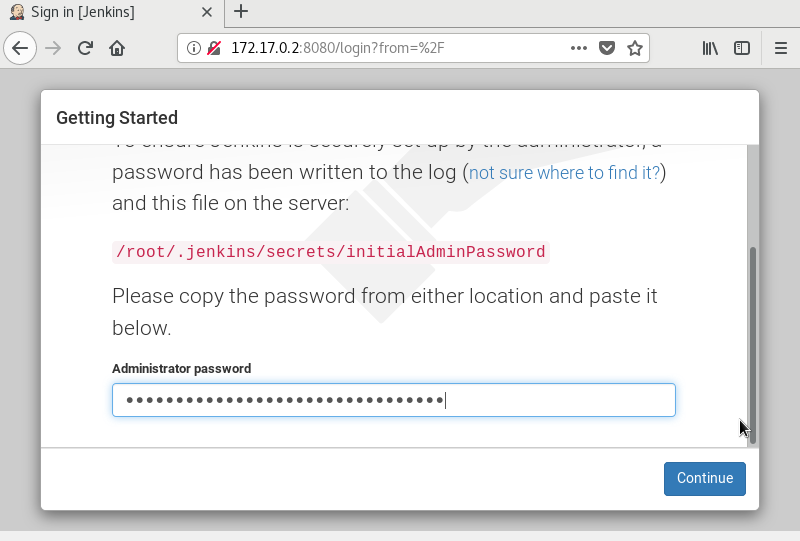


Build the image and launch it, when you launch the image it’ll automatically start Jenkins services.



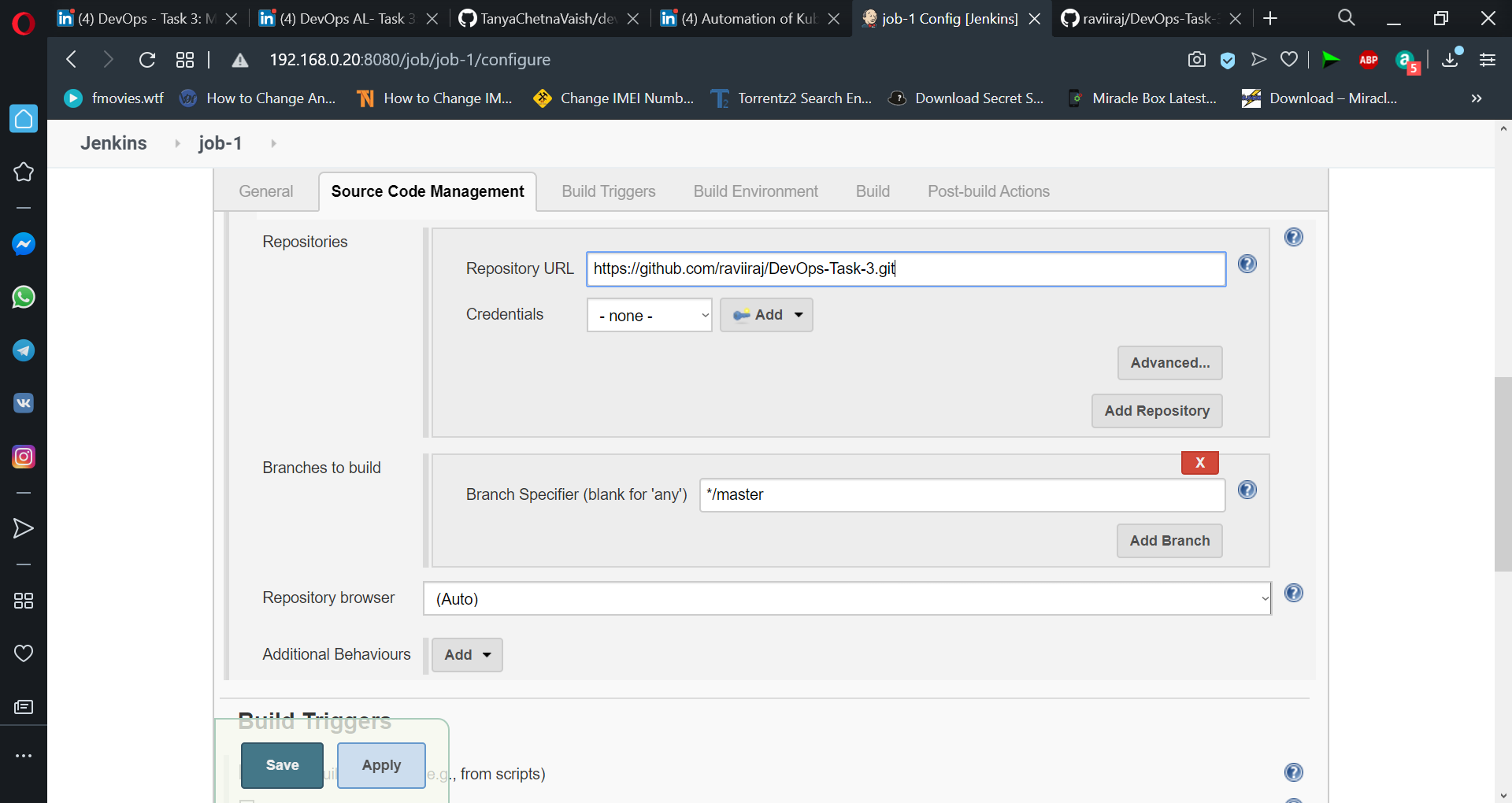
To know the IP of docker container

[[#Docker inspect container\_name]]



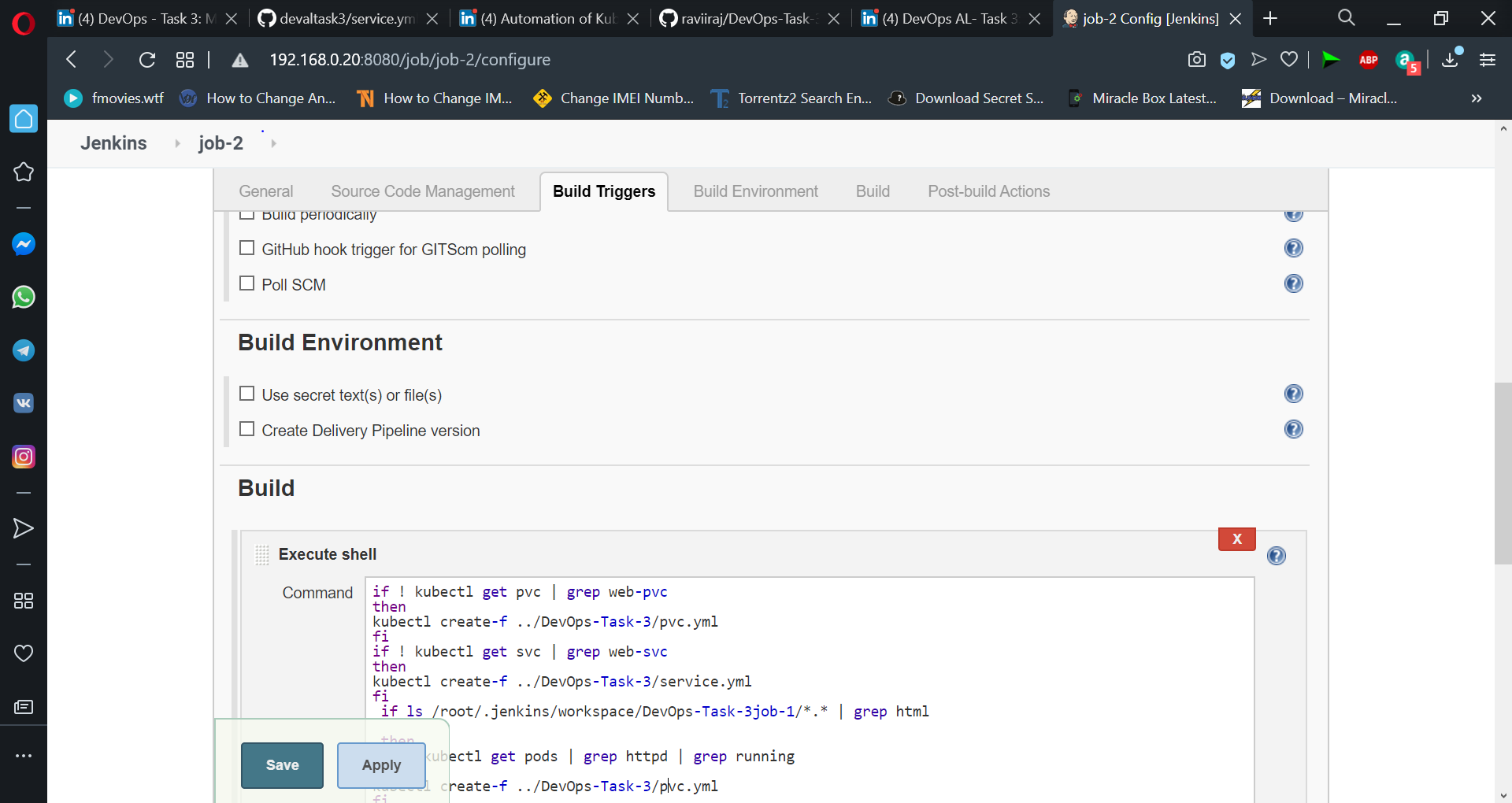
**Job-1**: Pull the GitHub repo automatically when some developers push the repo to GitHub.

To do Job-1 we have selected git in SCM and gave git URL. Then, selected GitHub hook trigger for SCM polling and copied pulled repository code in /host/web-repo.



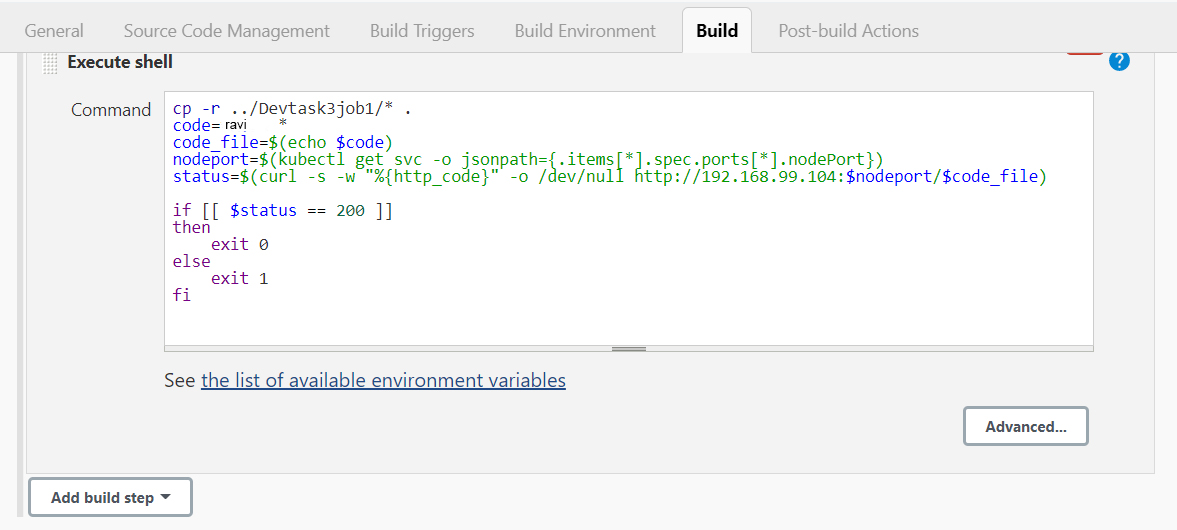
**Service.yml** file that will create a service. **pvc.yml** file which will create a Persistent Volume Claim, which keeps data persistent. It can be mounted to any container. PVC creates PV dynamically. **httpd.yml** and **php.yml** file which will launch pods for HTTP and PHP respectively, when run.

Job-2 creation and writing code to launch a service if the service isn't launched already. Launched a PVC. Also launch a pod, based on our code. If our code is .php then a PHP pod will be launched using a php.yml file. If the code is .html then HTTP pose will be launched using httpd.yml file, as provided.

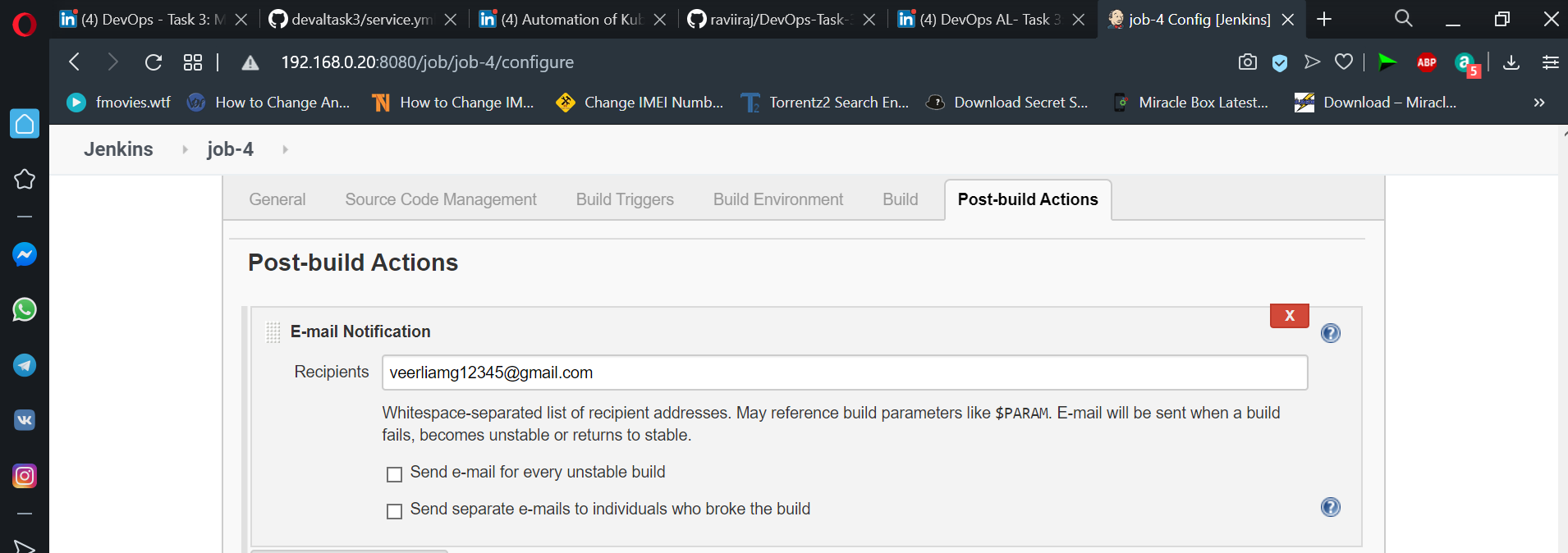


This job will trigger job3, which will check for if the code runs successfully or not.

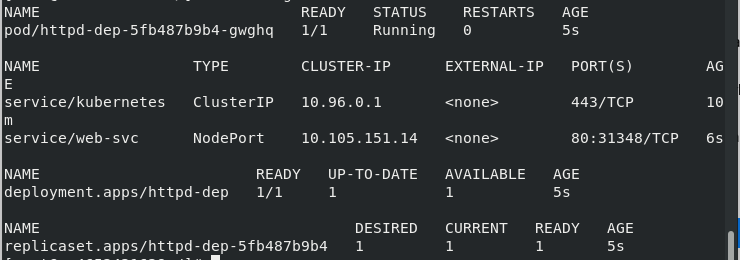
Job -4 for checking validity of codeby the status of the code. If the page returns code 200 that means it runs successfully, otherwise the build fails. Added parameterized trigger that builds other projects based on parameters specified. Here this job3 builds job4 only when it fails or is unstable.



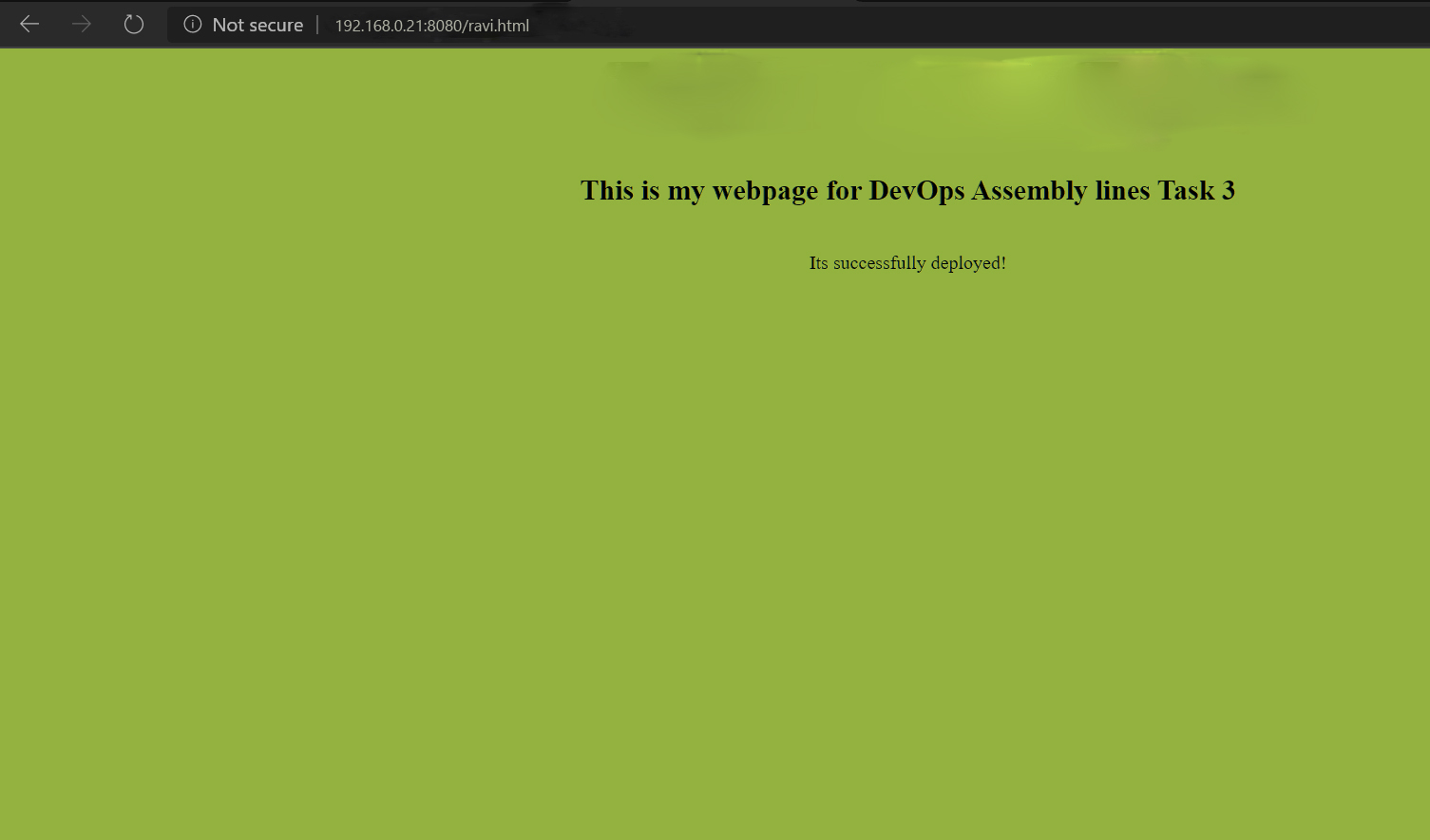
Finally job-4 will send email notification to the user when the app/webserver isn't working. For this Email plugin and Email-extension plugin must be installed which will help in system-generated emails, which can be customized. Here provide the recipient email address. Also write any customized message or text which you want to send in email, regarding the unsuccessful deployment of app/webserver.



Those file .html file that is ravi.html file, on successful built of job-1 and job-2 launch httpd pod service and PVC setup. On running kubectl get all command we can see out httpd pod deployed with our HTML code and exposed as well. Which can be checked using ip and port.

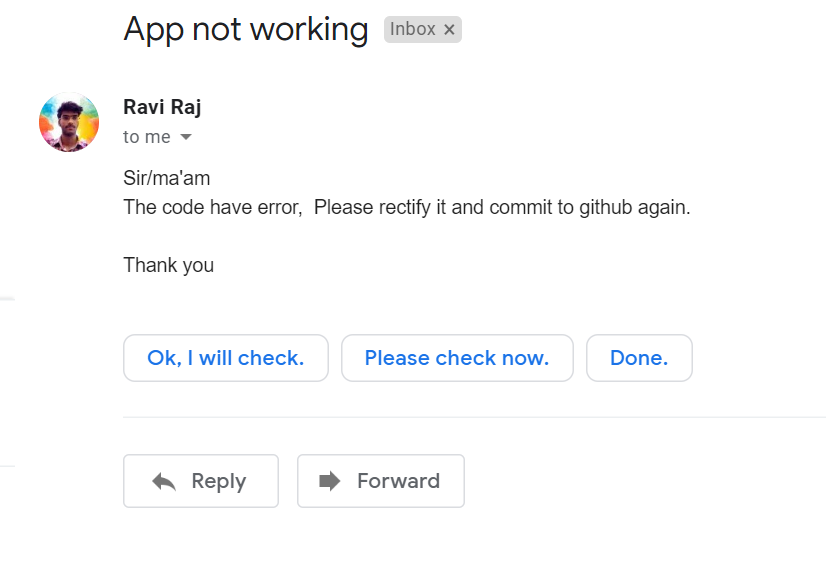


Build pipeline for html successful build.



Also I added a PHP file ravi.php. The job-1, job-2, job-3 will run successfully when the code is right. Job4 won't run, returning status 200.php pod will be launched for this. The site will be successfully deployed.

Due to wrong updation of PHP code, and commit this code to Github. It will trigger job1 and job2 successfully, but job3 will fail, since the code wasn't right returning to status other than 200. Hence job-4 will be triggered now since job3 fails. Thus the user will receive a message for the unsuccessful build.



This time Build pipeline will fail showing error, for futher code details you can checkout github account.

Thankfully I have completed this task.