

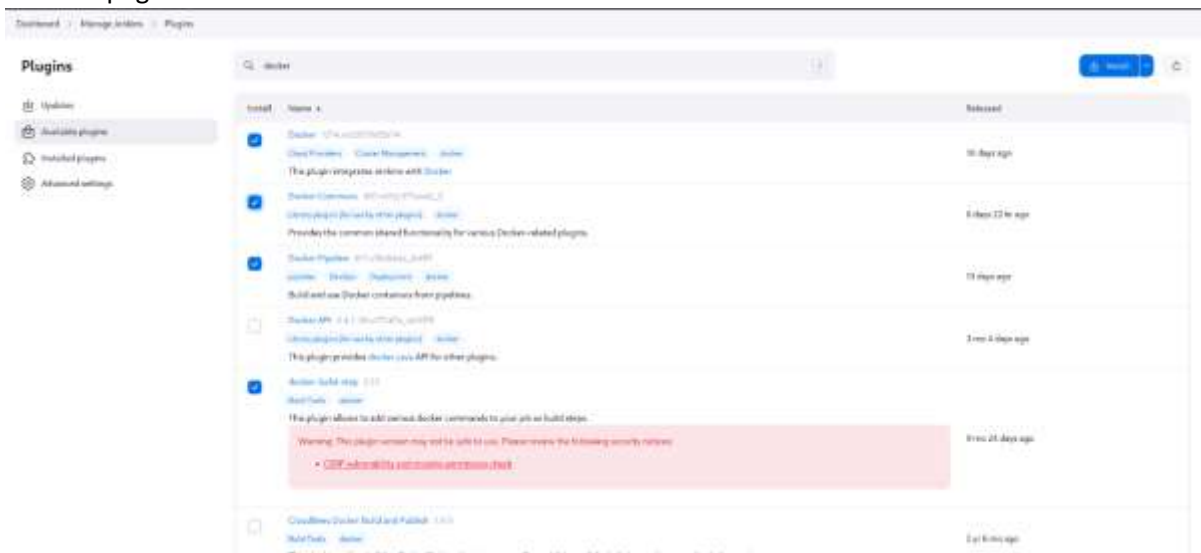
[illegible]

```
root@bathes:~# sudo apt install docker.io
Err:4 https://download.docker.com/linux/ubuntu noble InRelease
  Temporary failure resolving 'download.docker.com'
Ign:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:6 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:2 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Err:7 http://archive.ubuntu.com/ubuntu noble InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:5 http://archive.ubuntu.com/ubuntu noble-updates InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Err:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
  Temporary failure resolving 'archive.ubuntu.com'
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
28 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: Failed to fetch https://download.docker.com/linux/ubuntu/dists/noble/InRelease Temporary failure resolving 'download.docker.com'
W: Failed to fetch https://pkg.jenkins.io/debian-stable/binary/InRelease Temporary failure resolving 'pkg.jenkins.io'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/noble/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/noble-updates/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://archive.ubuntu.com/ubuntu/dists/noble-backports/InRelease Temporary failure resolving 'archive.ubuntu.com'
W: Failed to fetch http://security.ubuntu.com/ubuntu/dists/noble-security/InRelease Temporary failure resolving 'security.ubuntu.com'
W: Some index files failed to download. They have been ignored, or old ones used instead.
root@bathes:~# sudo apt install -y docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
 containerd.io : Conflicts: containerd
E: Error, pkgProblemResolver::Resolve generated breaks, this may be caused by held packages.
root@bathes:~#
```

Download Docker plugins in Jenkins

- Go to Jenkins `Dashboard` -> `Manage Jenkins` -> `Available Plugins` -> Search `Docker`
- Select these plugins and Install
 - Docker
 - Docker Commons
 - Docker Pipeline
 - docker-build-step
 - CoudBees Docker Build and Publish
- In this page Check the `



Restart Jenkins` after installation this will restart Jenkins

Plugins

- Upstream
- Available plugins
- Installed plugins
- Advanced settings

docker

1. Installed

Install	Name	Released
<input checked="" type="checkbox"/>	CloudBees Docker Build and Publish 1.0.1 Description Source This plugin enables building Dockerfile based projects, as well as publishing of the built images to the docker registry. Amazon ECR 1.0.0 (alpha, 1.0.0 beta, 1.0)	2 yr 8 mo ago
<input type="checkbox"/>	Docker This plugin generates Docker authentication token from Amazon Credentials to access Amazon ECR. This plugin is up for adoption into the Jenkins core ecosystem. View on GitHub View on Jenkins for more information.	1 year 10 days ago
<input type="checkbox"/>	Docker Commons Build Step 1.0 Docker Commons plugin for Jenkins.	6 yr 6 mo ago
<input type="checkbox"/>	Docker Client 1.0.0 Docker Uses Docker containers to run Jenkins build agents.	7 yr 7 mo ago
<input type="checkbox"/>	Jfrog 1.1.0 JST Download JST Download JST Download JST Download JST Download The Jenkins Jfrog Plugin allows for easy integration between Jenkins and the Jfrog Platform. This integration allows your build jobs to deploy artifacts and receive dependencies for and from Artifactory, and then have them linked to the build job that created them. It also allows you to scan your artifacts and builds with Jfrog Xray and distribute your software package to remote locations using Jfrog Distribution. This is all achieved by the plugin by wrapping Jfrog CLI. Any Jfrog CLI command can be executed from within your Jenkins Pipeline job using the Jfrog Plugin.	2 mo 5 days ago
<input type="checkbox"/>	CloudBees Docker Commons Build Environment 1.0.0 Docker This builds inside a docker container, defined by a Docker image or Dockerfile stored in project SCM.	7 yr 6 mo ago

Plugins

- Upstream
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Download progress

Progress

- Checking network connectivity
- Checking update center connectivity
- Success

Cloud Statistics	Success
Authentication Tokens API	Success
Broker Commons	Success
Apache HttpClient/Component Client 5.x API	Success
Commons Compress API	Success
Docker API	Success
Docker	Success
Docker Commons	Success
Docker Pipeline	Success
Jenkins	Success
Dev Tools Symbols API	Success
Git Dependency	Success
Google Integration	Success
Google Material View	Success
Google plugin extensions	Success

→ [Go back to the top page](#)

You can start using the installed plugins right away!

→ [Restart Jenkins when installation is complete and no jobs are running](#)

Jenkins 2.403.1



Jenkins is restarting

Your browser will reload automatically when Jenkins is ready.

Safe Restart

Builds are agents are usually continuous.

Add Jenkins to Docker group

- Go to terminal and run these commands to add Jenkins to docker group

```
``bash
```

```
sudo usermod -aG docker jenkins
```

```
sudo systemctl restart jenkins
```

```
sudo reboot
```

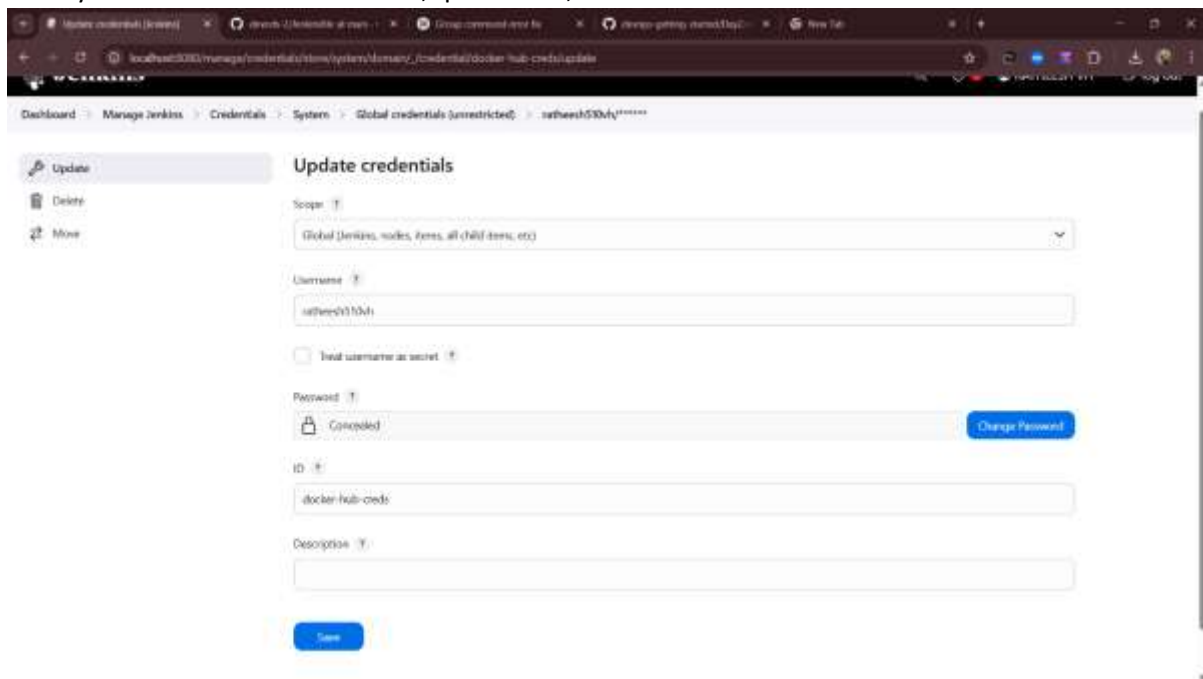
```
```
```

- This will do its thing and reboot the system

## ## Setting up docker credentials

- Go to Jenkins > `Manage Jenkins` > `Credentials` > `System` > `Global Credentials (Unrestricted)` > `Add Credentials`

- Fill your Docker hub `username`, `password`, and in the `id` field enter `docker-seccred`



The screenshot shows the Jenkins web interface for updating credentials. The breadcrumb trail is: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) > safford550dy\*\*\*\*\*. The form is titled 'Update credentials' and includes the following fields and options:

- Scope:** A dropdown menu set to 'Global (Jenkins, nodes, zones, all child items, etc)'.
- Username:** A text input field containing 'safford550dy'.
- Treat username as secret:** An unchecked checkbox.
- Password:** A text input field containing 'Corrupted', with a 'Change Password' button to its right.
- ID:** A text input field containing 'docker-hub-creds'.
- Description:** An empty text input field.
- Buttons:** 'Update' (disabled), 'Delete', 'Move', and a blue 'Save' button at the bottom.

## ## Creating and building a pipeline

- Go to Jenkins `Dashboard` > `Create a Job`

- Enter a project name

- Select `pipeline`

- Click `Ok`

- Go to `pipeline`

- Paste this script below and change the credential wherever mentioned:

```
``groovy
pipeline {
 agent any

 environment {
 IMAGE_NAME = "sivagurunathan7/test_repository"
 TAG = "latest"
 CONTAINER_NAME = "my-container"
 PORT = "3001"
 }

 stages {

 stage('Clone Repository') {
 steps {
 echo "Cloning GitHub repository..."
 git 'https://github.com/Sivagurunathan98/New_docker_push.git'
 }
 }
 }
}
```

```
stage('Build Docker Image') {
```

```
 steps {
```

```
 echo "Building Docker image..."
```

```
 bat 'powershell.exe -Command "Set-ExecutionPolicy RemoteSigned -Force"'
```

```
 bat 'powershell.exe -File .\\Build.ps1'
```

```
 }
```

```
}
```

```
stage('Login to Docker Hub') {
```

```
 steps {
```

```
 echo "Logging into Docker Hub..."
```

```
 withCredentials([usernamePassword(credentialsId: 'docker-hub-creds', usernameVariable: 'DOCKER_USER', passwordVariable: 'DOCKER_PASS')]) {
```

```
 bat 'echo %DOCKER_PASS% | docker login -u %DOCKER_USER% --password-stdin'
```

```
 }
```

```
 }
```

```
}
```

```
stage('Push Docker Image') {
```

```
 steps {
```

```
 echo "Pushing Docker image to Docker Hub..."
```

```
 bat "docker tag %IMAGE_NAME%:%TAG% %IMAGE_NAME%:%TAG%"
```

```
 bat "docker push %IMAGE_NAME%:%TAG%"
```

```
 }
```

```
}
```

```
stage('Deploy Docker Container') {
```

```
 steps {
```

```
 echo "Deploying Docker container..."
```

```
 bat 'powershell.exe -File .\\Deploy.ps1'
```

```
 }
```

```

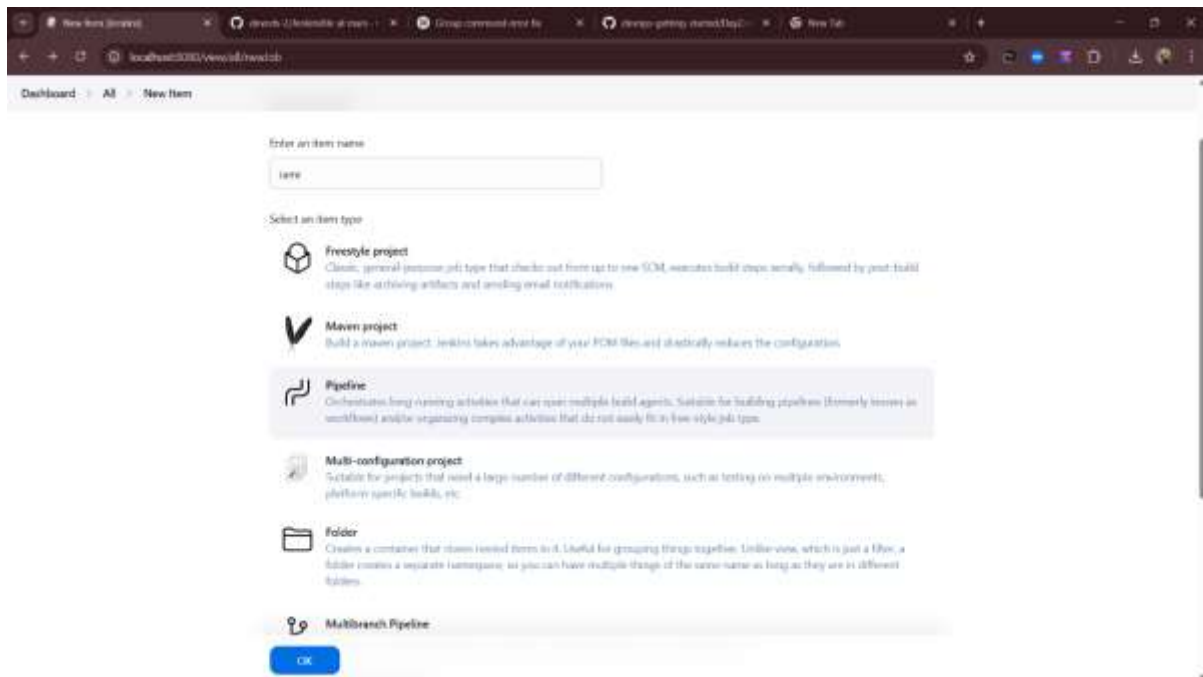
 }
}

post {
 success {
 echo "Deployment Successful!"
 }
 failure {
 echo "Deployment Failed!"
 }
}
}
'''

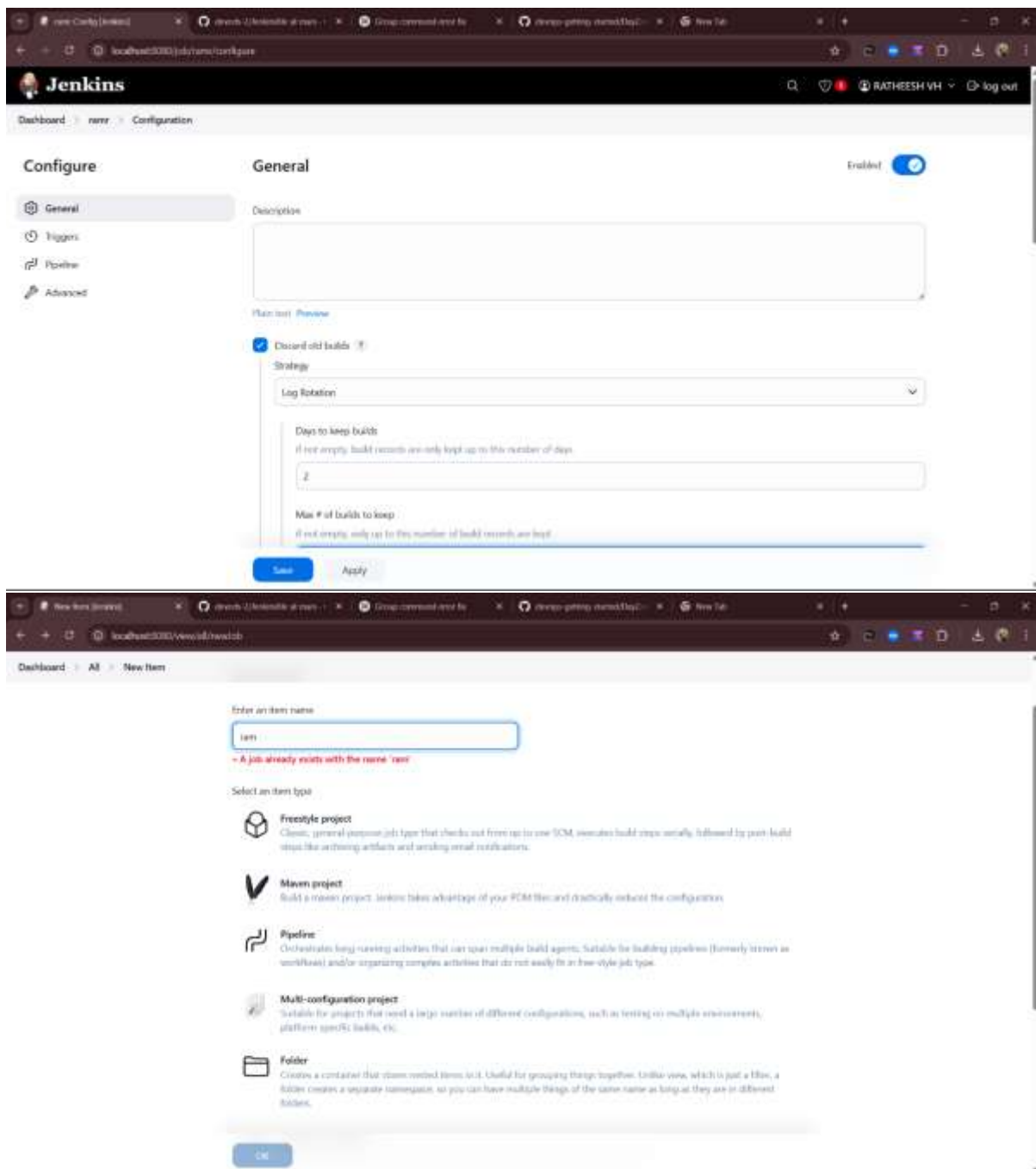
```

- click `save`

- click `build`







- Go to Docker Hub to see your image pushed there