Jenkins Building Docker Image and Sending to Registry

Gustavo Apolinario · Follow
7 min read · May 5, 2018

Listen Share

Hi everyone. In this tutorial we will create a docker image with jenkins and send then to dockerhub.

What's docker?
Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications.

You can learn more about docker in docker website

Why pipeline?
You can reuse everything you did, put your jenkins code inside git project, the change in pipeline is showed in "changes" inside job history

What's dockerhub?
Dockerhub is a public docker registry to store your docker images inside. If you want a private registry, you can pay for it. We will use it because it is the most easeful docker registry.

What's docker registy?
Docker registry is a server to distribute versions of docker images.

Jenkins with docker installed

We will use some pipeline codes, the jenkins need have installed docker inside him to find this commands.

I created a docker image of jenkins with docker installed.

Let's see the Dockerfile:

```
FROM jenkins/jenkins:lts

USER root

RUN apt-get update && \
apt-get -y install apt-transport-https \
    ca-certificates \
    curl \
    gnupg2 \
    software-properties-common && \
curl -fsSL https://download.docker.com/linux/$(. /etc/os-release;
echo "$ID")/gpg > /tmp/dkey; apt-key add /tmp/dkey && \
add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/$(. /etc/os-
release; echo "$ID") \
    $(lsb_release -cs) \
    stable" && \
apt-get update && \
apt-get -y install docker-ce

RUN apt-get install -y docker-ce

RUN usermod -a -G docker jenkins

USER jenkins
```

This Dockerfile is builded from jenkins official image, install docker and give access to user jenkins build dockers.

You can build this image, but is already in dockerhub gustavoapolinario/jenkins-docker.

Running jenkins with docker from host

To run the container, you need add a volume in docker run command.

```
… -v /var/run/docker.sock:/var/run/docker.sock …
```

It will share the docker socket (used in your machine) with the container.

The complete run command:

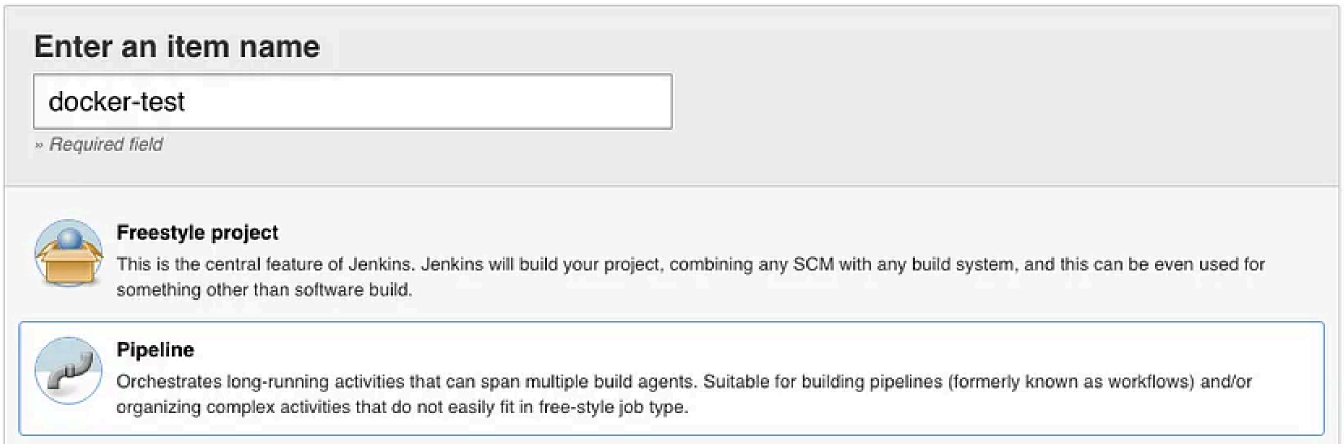
```
docker run --name jenkins-docker -p 8080:8080 -v
/var/run/docker.sock:/var/run/docker.sock gustavoapolinario/jenkins-
docker
```

After inicialized the jenkins, complete the jenkins startup wizard.

Read this tutorial to complete wizard and install the locale and blueocean plugins: Quick start with jenkins in docker.

Creating a job to test docker command

In home of jenkins, click on “New Item”, select “Pipeline” and put the job name as “docker-test”.



Enter an item name

docker-test

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

New pipeline Job

Put this script inside of job:

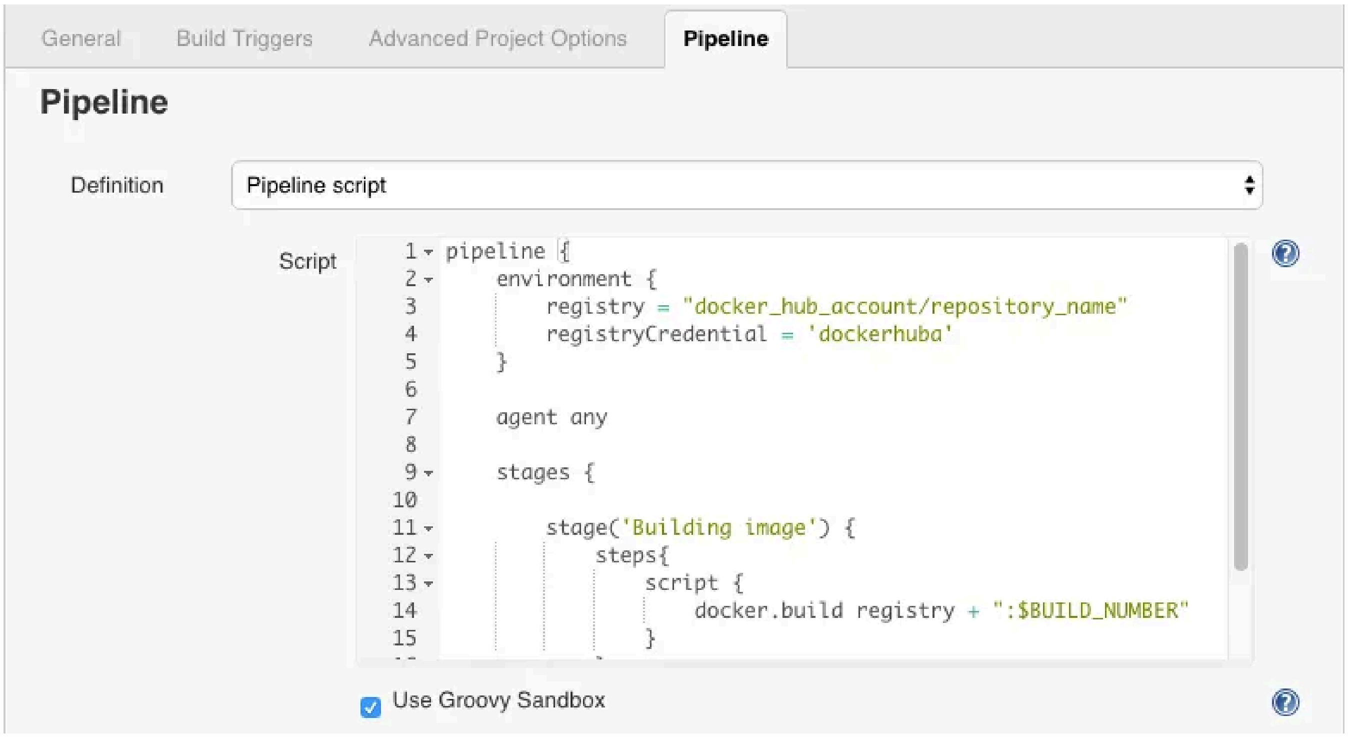
```
pipeline {

  environment {
    registry = "docker_hub_account/repository_name"
    registryCredential = 'dockerhub'
  }

  agent any

  stages {
    stage('Building image') {
      steps{
        script {
          docker.build registry + ":$BUILD_NUMBER"
        }
      }
    }
    }
}
```

The screen will be like this:



Pipeline

Definition: Pipeline script

Script

```
pipeline {
    environment {
        registry = "docker_hub_account/repository_name"
        registryCredential = 'dockerhuba'
    }

    agent any

    stages {

        stage('Building image') {
            steps{
                script {
                    docker.build registry + ":$BUILD_NUMBER"
                }
```

[✓] Use Groovy Sandbox

Pipeline in job config

Save the job.

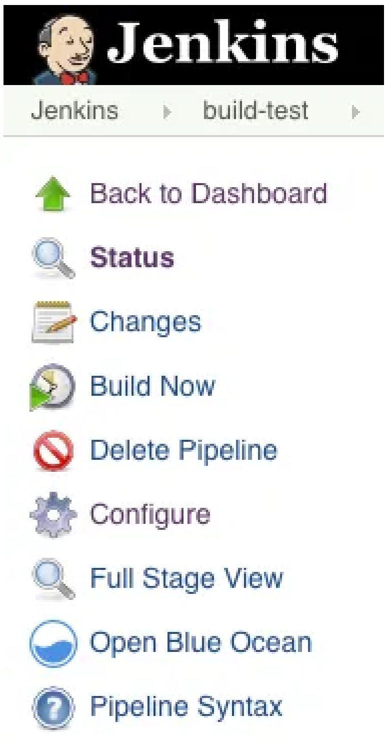
```
environment {
    registry = "docker_hub_account/repository_name"
    registryCredential = 'dockerhub'
}
```

The job will have one step. It will run the docker build and use the jenkins build number in docker tag. With build number turn easeful to deploy or rollback based in jenkins.

```
stages {
  stage('Building image') {
    steps{
      script {
        docker.build registry + ":$BUILD_NUMBER"
      }
    }
  }
}
```

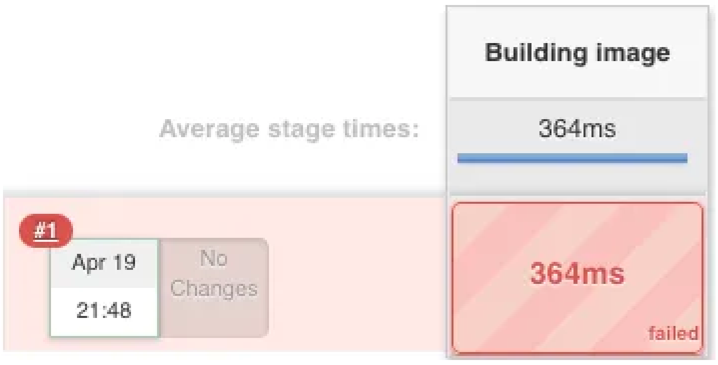
Testing the docker command in job

Click on "Build Now" in job's menu.



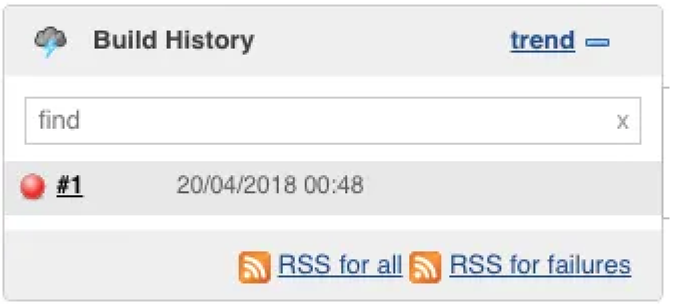
Job Menu

The job will failure. Don't worry.



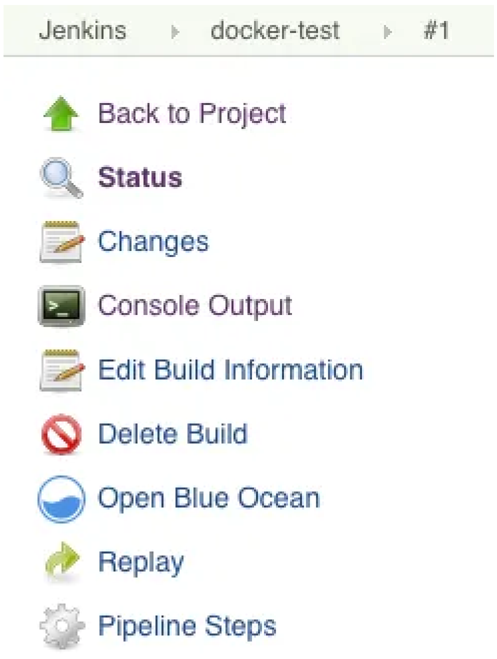
Build Failure

In job's home, you can click in circle and see the console output.



Build History

Or click on build number (#1) and click on "Console Output".



Job Build Menu

In Console Output you will see this:

```
Started by user GUSTAVO WILLY APOLINARIO DOMINGUES
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/docker-test
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Building image)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
[teste234] Running shell script
+ docker build -t docker_hub_account/repository_name:1 .
unable to prepare context: unable to evaluate symlinks in Dockerfile
path: lstat /var/jenkins_home/workspace/docker-test/Dockerfile: no
such file or directory
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
```

The error happens because the Dockerfile is not finded. We will resolve it soon.

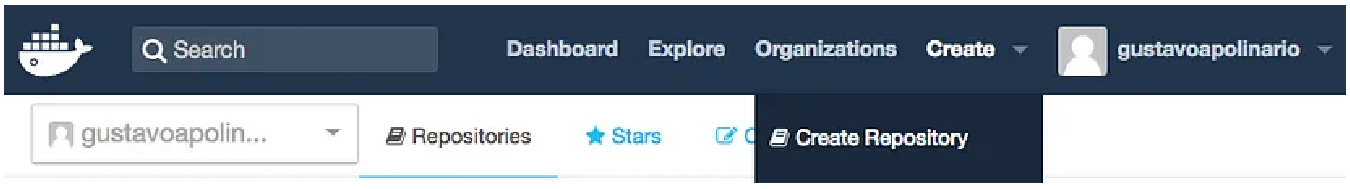
The docker command is executed and the jenkins found the command.

Creating a dockerhub repository

Create a dockerhub account, if you don't have yet.

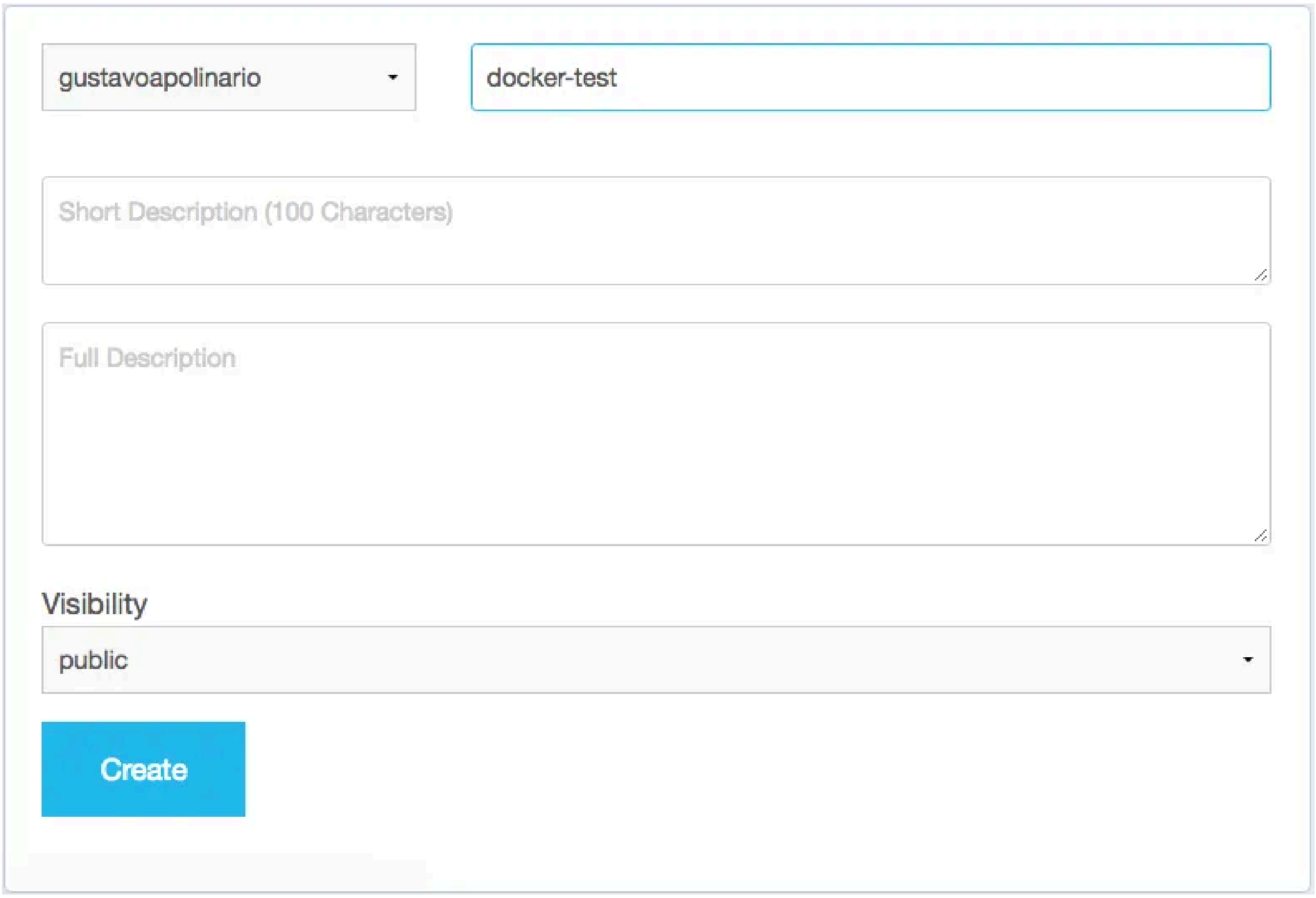
dockerhub

Logged in dockerhub, click on "Create" > "Create Repository".



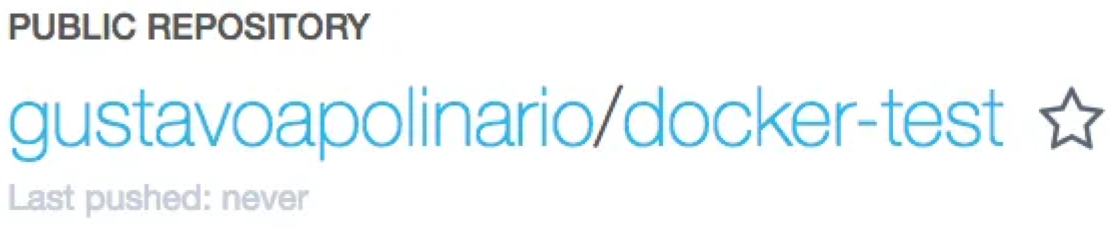
Dockerhub menu to Create Repository

Put a name for your repository. For this example, use "docker-test".



Creating Dockerhub Repository

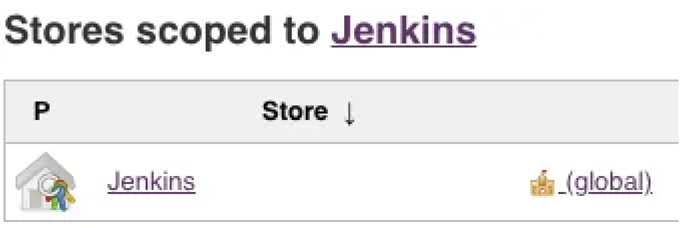
After docker repository created, get the name to use in our pipeline. In my case, the name is “**gustavoapolinario/docker-test**”.



Dockerhub Repository

Creating dockerhub credential

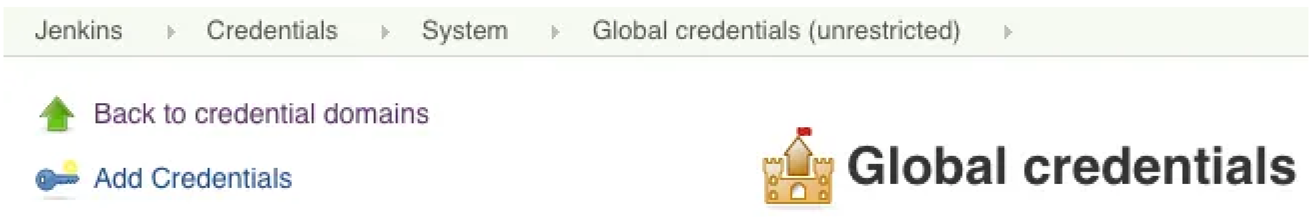
Go to jenkins home, click on “credentials” and “**(global)**”.



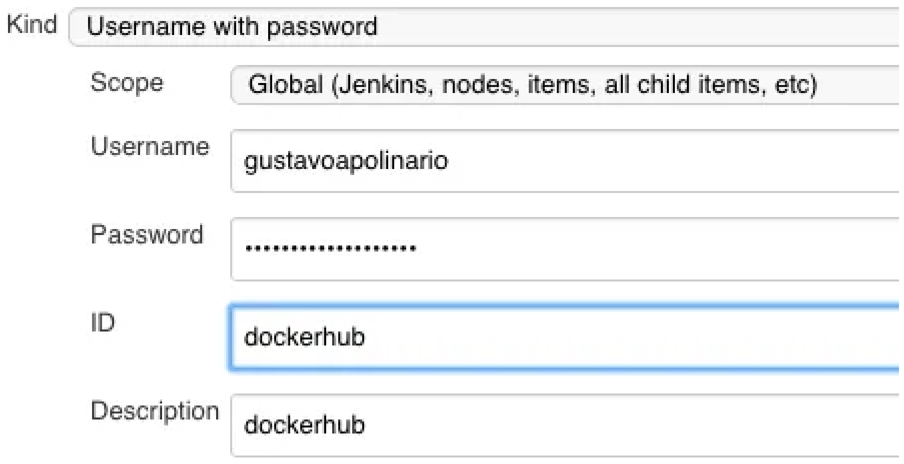
P	Store ↓	
	Jenkins	(global)

Credentials

Click on "Add Credentials" in left menu.



Put your credential and save it.



Kind	Username with password
Scope	Global (Jenkins, nodes, items, all child items, etc)
Username	gustavoapolinario
Password	•••••••••••••••••••
ID	dockerhub
Description	dockerhub

Remenber to change the credential environment (registryCredential) if you didn't put "**dockerhub**" in Credential ID.

The credential is configured.

Configuring the environment of dockerhub

Alter the job pipeline. Go to jenkins home, click on job name (docker-test), click on "Configure" in job menu.

The code you need to change is:

```
environment {
    registry = "docker_hub_account/repository_name"
    registryCredential = ‘dockerhub’
}
```

Change the environment variable "registry" to your repository name. In my case is "**gustavoapolinario/docker-test**".

Change the environment variable "registryCredential" if necessary.

My environment is:

```
environment {
    registry = "gustavoapolinario/docker-test"
    registryCredential = ‘dockerhub’
}
```

Building the first docker image

With dockerhub credential and repository created, the jenkins can send the docker image builded to dockerhub (our docker repository).

In this example, let's build a node.js application. We need a Dockerfile to the build.

Let's create a new step in pipeline to clone a git repository that have a Dockerfile inside.

```
stage('Cloning Git') {
  steps {
    git 'https://github.com/gustavoapolinario/microservices-node-
example-todo-frontend.git'
  }
}
```

The pipeline must be (but using your environment):

```
pipeline {
  environment {
    registry = "gustavoapolinario/docker-test"
    registryCredential = ‘dockerhub’
  }
  agent any
  stages {
    stage('Cloning Git') {
      steps {
        git 'https://github.com/gustavoapolinario/microservices-
node-example-todo-frontend.git'
      }
    }
    stage('Building image') {
      steps{
        script {
          docker.build registry + ":$BUILD_NUMBER"
        }
      }
    }
  }
}
```

Save and run it clicking on “Build Now”

The Stage view in jenkins job will change to this:

Deploying the docker image to dockerhub

At this moment, we clone a git and build a docker image.

We need put this image in docker registry to pull it in other machines.

First, create a environment to save docker image informations.

```
dockerImage = ''
```

Change the build stage to save build information in environment.

```
dockerImage = docker.build registry + ":$BUILD_NUMBER"
```

Ceate a new stage to push the docker image builded to dockerhub.

```
stage('Deploy Image') {
  steps{

    script {
      docker.withRegistry( '', registryCredential ) {
        dockerImage.push()
      }
    }
  }
}
```

After build and deploy, delete the image to cleanup your server space.

```
stage('Remove Unused docker image') {
  steps{
    sh "docker rmi $registry:$BUILD_NUMBER"
  }
}
```

The final code will be (remember, using your environment):

```
pipeline {
  environment {
    registry = "gustavoapolinario/docker-test"
    registryCredential = 'dockerhub'
    dockerImage = ''
  }
  agent any
  stages {
    stage('Cloning Git') {
      steps {
        git 'https://github.com/gustavoapolinario/microservices-
node-example-todo-frontend.git'
      }
    }
    stage('Building image') {
      steps{
        script {
          dockerImage = docker.build registry + ":$BUILD_NUMBER"
        }
      }
    }
    stage('Deploy Image') {
      steps{
        script {
          docker.withRegistry( '', registryCredential ) {
            dockerImage.push()
          }
        }
      }
    }
    stage('Remove Unused docker image') {
      steps{
        sh "docker rmi $registry:$BUILD_NUMBER"
      }
    }
  }
}
```

Save and run it.

Awesome, the build is complete and the image will be send to docker registry.

Complete pipeline to a node.js application

This step is for who did this tutorial: Jenkins Starting with Pipeline doing a Node.js test.

To complete the node.js pipeline, let's change this pipeline to merging with node.js build and test.

The complete pipeline will be:

```
pipeline {
  environment {
    registry = "gustavoapolinario/docker-test"
    registryCredential = 'dockerhub'
    dockerImage = ''
  }
  agent any
  tools {nodejs "node" }
  stages {
    stage('Cloning Git') {
      steps {
        git 'https://github.com/gustavoapolinario/node-todo-
frontend'
      }
    }
    stage('Build') {
       steps {
         sh 'npm install'
       }
    }
    stage('Test') {
      steps {
        sh 'npm test'
      }
    }
    stage('Building image') {
      steps{
        script {
          dockerImage = docker.build registry + ":$BUILD_NUMBER"
        }
      }
    }
    stage('Deploy Image') {
      steps{
         script {
            docker.withRegistry( '', registryCredential ) {
            dockerImage.push()
          }
        }
      }
    }
    stage('Remove Unused docker image') {
      steps{
        sh "docker rmi $registry:$BUILD_NUMBER"
      }
    }
  }
}
```

Now you have a pipeline for test and create a docker image for your application.

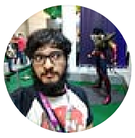
Thanks.

— — — — — — — — — — — — — — — —

A special thanks to Matthias Döring, who notify me about the lack of docker cleanup. It is necessary to your server don't use all disc space. The docker images are usefull bexause they are cache to next build, but use a lot of disc space.

https://medium.com/@cryptolukas/you-should-add-this-as-last-stage-or-post-task-d69fb384a361

Docker Jenkins Docker Registry Dockerhub Pipeline



Follow

Written by Gustavo Apolinario

377 Followers · 11 Following

I'm a FullStack Developer, working now with DevOps automations. I previously worked as a front-end, PHP Developer, and DevOps.

Responses (22)

What are your thoughts?

Respond

Matthias Döring
almost 6 years ago

You should add this as last stage or post task

stage('Docker Purge') {

steps {

sh 'docker image prune -fa'

deleteDir()

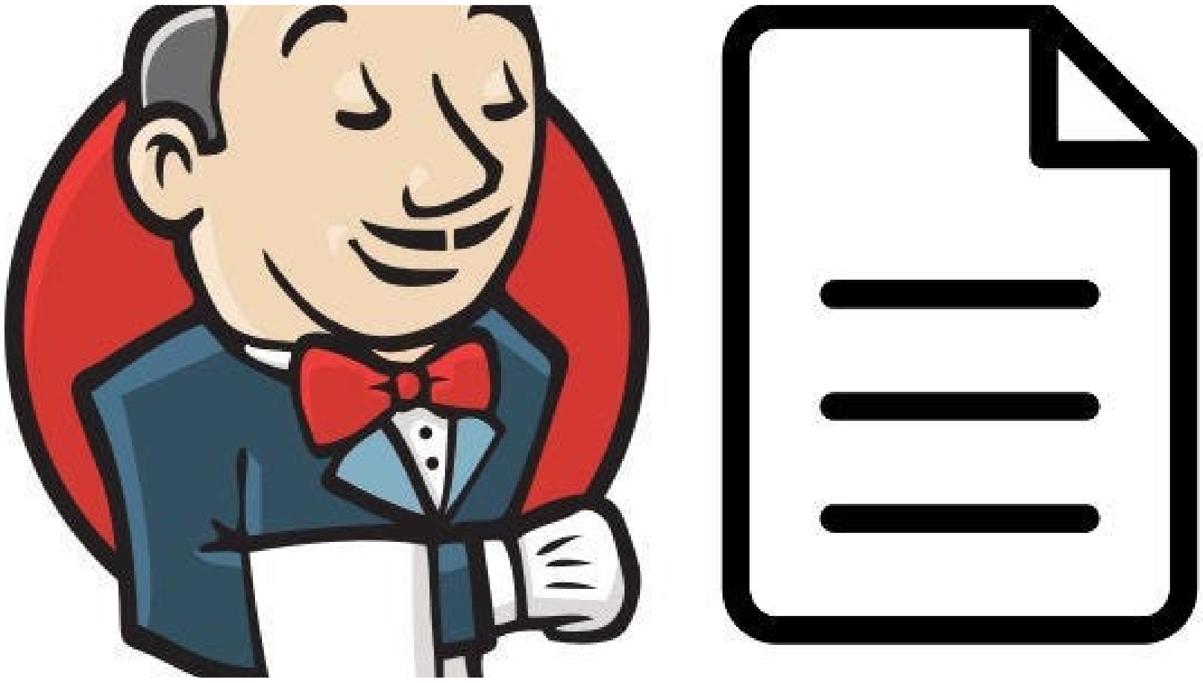
}...

Read More

37 1 reply Reply

See all responses

More from Gustavo Apolinario

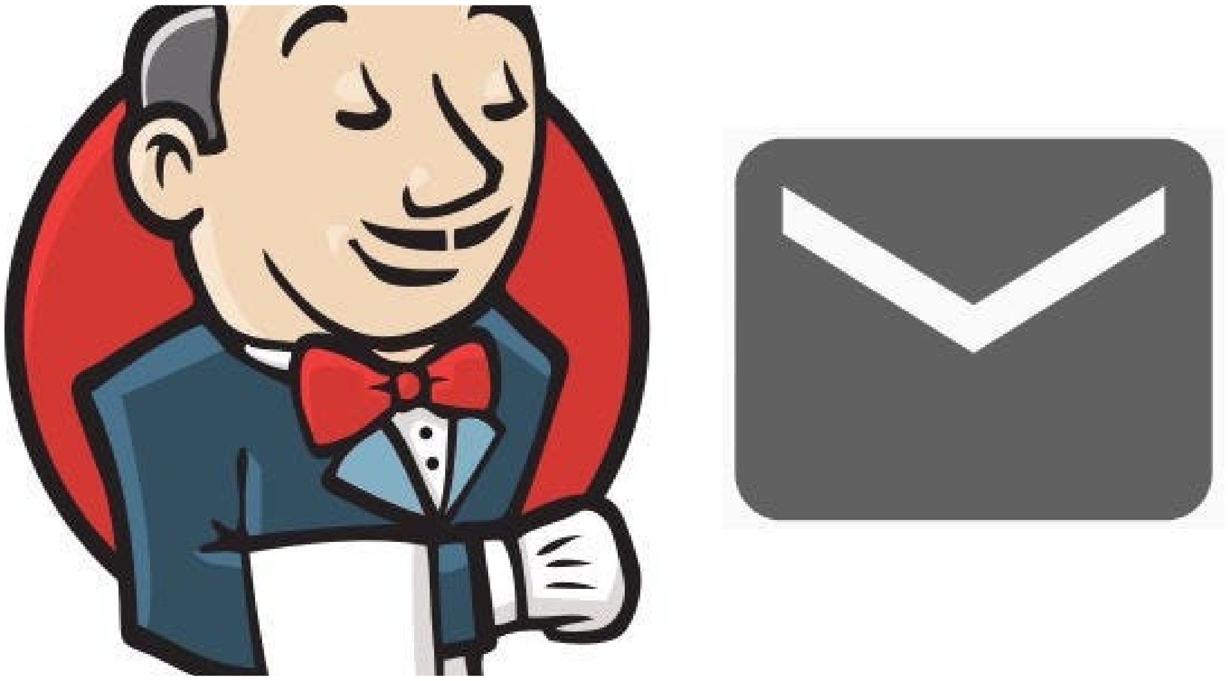


Gustavo Apolinario

Jenkins archive artifact/save file in Pipeline

In this tutorial i will show how to save files “outside build workspace” to get after others builds.

Aug 31, 2018 182 6



Gustavo Apolinario

Jenkins send email with attachment

This tutorial is a complement for my other tutorial: Jenkins sending email on post build.

Aug 31, 2018 39 5

Gustavo Apolinario

Jenkins sending email on post build

After Build you need communicate the persons who can be responsables for the build failure.

May 14, 2018 501 16

Gustavo Apolinario

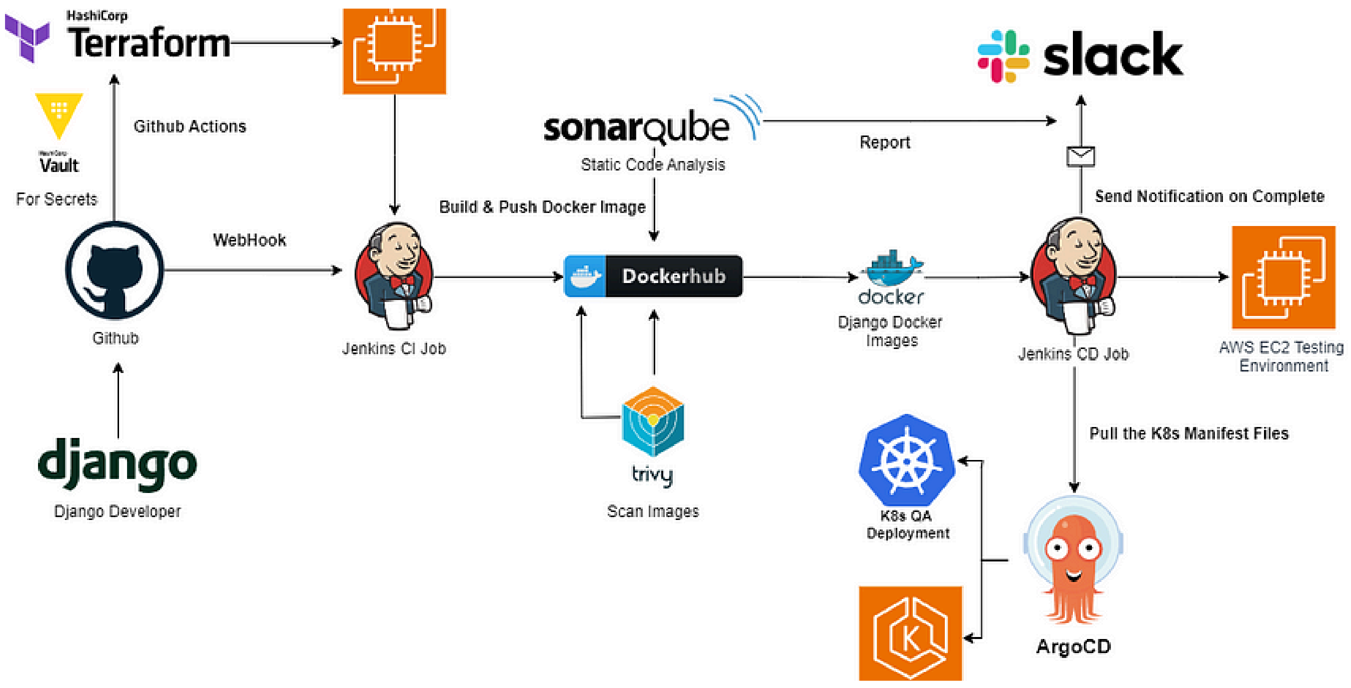
How to do Post Build in Jenkins Pipeline

It’s a sample of how to do post build using pipeline inside jenkins.

May 11, 2018 172 3

See all from Gustavo Apolinario

Recommended from Medium



In Django Unleashed by Joel Wembo

Technical Guide: End-to-End CI/CD DevOps with Jenkins, Docker, Kubernetes, ArgoCD, Github Actions ...

Building an end-to-end CI/CD pipeline for Django applications using Jenkins, Docker, Kubernetes, ArgoCD, AWS EKS, AWS EC2

Apr 12 960 20

Harendra

How I Am Using a Lifetime 100% Free Server

Get a server with 24 GB RAM + 4 CPU + 200 GB Storage + Always Free

Oct 26 7K 106

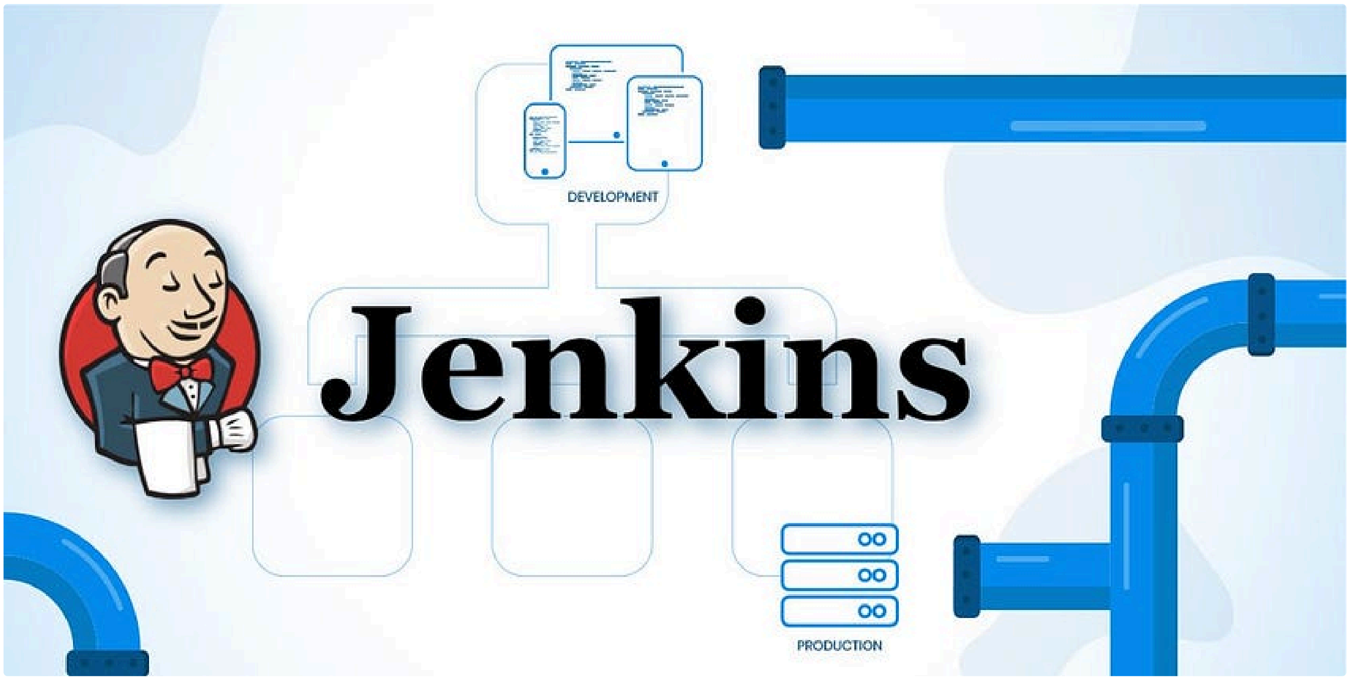
Lists

Coding & Development

11 stories · 934 saves

Natural Language Processing

1853 stories · 1478 saves



Kuldeepkumawat

How To Set Jenkins Pipeline Environment Variables?

Introduction

Aug 26

In devsecops-community by Karthick Dkk

Day 28: Why Jenkins? An Overview of CI/CD

An overview of CI/CD using Jenkins for beginners

Oct 18 6

Jo Wang

Docker

Docker simplifies the process of building, distributing, and running applications. It achieves this through containerization, which allows...

Jun 26

Idrak Mirzayev

CI/CD with Jenkins

Jenkins is an open-source automation server that enables developers to build, test, and deploy their software in a streamlined, automated...

Sep 19

See more recommendations