

Dockerizing Jenkins Pipeline

Step 1: installation of docker

- Uninstall old versions

Older versions of Docker were called docker, docker.io, or docker-engine. If these are installed, uninstall them:

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
```

- Installation of docker

- SET UP THE REPOSITORY

Update the apt package index and install packages to allow apt to use a repository over HTTPS:

1. \$ sudo apt-get update
2. \$ sudo apt-get install \
3. apt-transport-https \
4. ca-certificates \
5. curl \
6. gnupg-agent \
7. software-properties-common

- Add Docker's official GPG key:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

- Set up the **stable** repository

```
$ sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) \
stable"
```

- o INSTALL DOCKER ENGINE

```
$ sudo apt-get update  
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

STEP 2: Setup a code workspace and the Github repository.

- 1.) Open terminal in Ubuntu server.
- 2.) Create a directory “Dockerizing-Jenkins-Pipeline” in the terminal and change directory.
- 3.) Run “git init” to initialize repository.
- 4.) Create a repository “ Dockerizing-Jenkins-Pipeline ” in the github.com
- 5.) Create a file name “README.md” add details of the project in the file.
- 6.) Execute the steps to do Initial commit, This will add the project in the github master branch.
 - a. git remote add origin https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
 - b. git add .
 - c. git commit -m “ Initial Commit”
 - d. git push -u origin master

STEP 3: Setup a Jenkins Server

1. Adding repos
`wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -`
2. Add below entry in sudo vi /etc/apt/sources.list
`deb https://pkg.jenkins.io/debian-stable binary/`
3. Update your local package index.
`sudo apt-get update`
4. Make sure you have jdk installed. If not, install it using the following command
`sudo apt-get install openjdk-8-jdk`
5. then install Jenkins
`sudo apt-get install Jenkins`
6. Select “Install Selected packages” and wait for installation to finish.
7. Create a new user after installation is finished.

Creating Jenkins integration with github

- A. Create a pipeline project with name “Docker jenkins intergration pipeline project” in Jenkins and under pipeline script ->pipeline syntax
- B. Select git as option -> then give git repo url -> username and password>
- C. Then generate the pipeline script
- D. Copy the script

```
git credentialsId: 'gituser', url:  
'https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git'
```
- E. Add that in pipeline steps

```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
                git credentialsId: 'gituser', url:
'https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git'
            }
        }
    }
}
```

- F. Now test the integration with build now.

 - 1) From your local Ubuntu system run below git command to push test file.
 - 2) echo "# Dockerizing-Jenkins-Pipeline" >> README.md
 - 3) git init
 - 4) git add README.md
 - 5) git commit -m "first commit"
 - 6) git remote add origin https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
 - 7) git push -u origin master
 - 8) Then now run the build now to test integration.
 - 9) We will see successful message as
 Git is successfully integrated with we can see test file in Jenkins work space.

- G. Now we have to make Jenkins to trigger job when git commit happens.

 1. Now we have to make Jenkins to trigger job when git commit happens.

2. For that, we have to add Jenkins url in git repository as webhook.
3. Settings → select Webhooks -> add Webhook -> "url of Jenkins"
4. Since in our project Jenkins running on private IP we can't use this url for webhook.
5. We are using Webhook Relay to forward our IP as mentioned below.

<https://webhookrelay.com/blog/2017/11/23/github-jenkins-guide/>

6. <https://my.webhookrelay.com/> Where from webhookrelay console we have to generate the new token like below in box.
7. Run the those commands on Ubuntu server where Jenkins installed.
8. Now we will get it will start forwarding localhost to below
<https://o0iwkgx6phxnun8wfpcej.hooks.webhookrelay.com>
9. Use the same url as webhook in github.
10. In Jenkins pipeline job select tick mark on GitHub hook trigger for GITScm polling

```
relay login -k e1ab8a62-675e-4671-91e7-019ce5afe85e -s jlzuSpJemQFk
export RELAY_KEY=e1ab8a62-675e-4671-91e7-019ce5afe85e
export RELAY_SECRET=jlzuSpJemQFk
relay forward --bucket github-jenkins http://localhost:8080/github-webhook/
```

STEP 4: Register and open <https://hub.docker.com/> with your own login.

- 1.) Create a new Docker repository named 'dockerizing_jenkins_pipeline'
- 2.) On the local directory Dockerizing-Jenkins-Pipeline create file with name 'Dockerfile' and add following contents and save the file.

```
FROM ubuntu:18.04
RUN apt-get update
RUN apt-get install -y apache2
RUN apt-get install -y apache2-utils
RUN apt-get clean
EXPOSE 80
copy login.html /var/www/html/index.html
CMD ["apache2ctl", "-D", "FOREGROUND"]
```

3) Now on the same directly create sample index.html file.

4) now on the Jenkins project edit the configure and go to pipeline and add below line mentioned in box, save and apply.

5) First time run the run below command to check whether image file is having any error or not.

```
docker build -t webserver:1.0
```

6) Finally we have add the Jenkins user to the Docker group to give permission to run the Docker command with command “ **usermod -a -G docker Jenkins** ”

```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                echo 'Hello World'
                git credentialsId: 'gituser', url:
'https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git'
            }
        }
        stage('Dockerbuild') {
            steps {
                sh '''
                    docker build -t webserver:1.0 .
                    docker tag webserver:1.0
                    vasanth95/dockerizing_jenkins_pipeline:webserver
                    docker login -u username -p password
                    docker push
                    vasanth95/dockerizing_jenkins_pipeline:webserver
                    docker run -d -p --name=httpserver 80:81
                    vasanth95/dockerizing_jenkins_pipeline:webserver
                    ...
                '''
            }
        }
    }
}
```

}

Testing the Docker Jenkins integration project.

Run below git command to add file to remote repository.

1. git add login.html
2. git add Dockerfile
3. git status
4. git commit -m "commiting change"
5. git push -u origin master

Now you we can see Jenkins job is triggered as soon as we commit the code.

Now Jenkins will do following jobs when we commit the

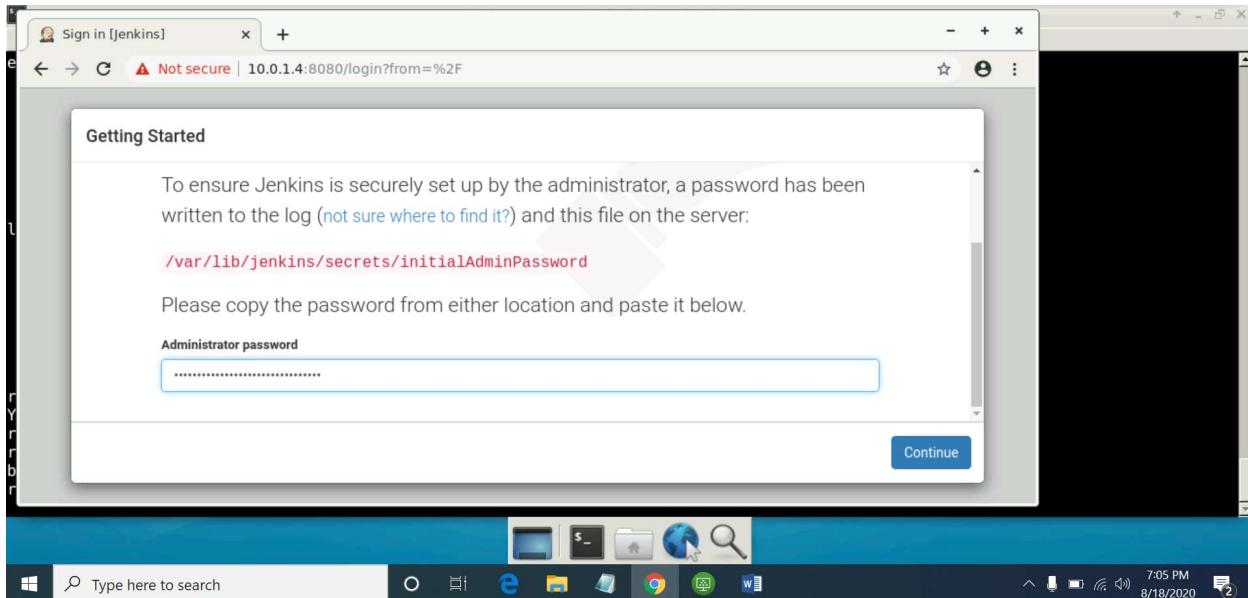
There are below stages in the Jenkins.

- a. Building Image
- b. Deploying the image in the dockerhub repository
- c. Executing Image from dockerhub.

```
root@vasanthakmindtr:~# apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.47.0-1ubuntu2.15).
gnupg-agent is already the newest version (2.1.11-6ubuntu2.1).
The following packages were automatically installed and are no longer required:
  aufs-tools cgroupfs-mount pigz
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  python3-software-properties
The following packages will be upgraded:
  apt-transport-https ca-certificates python3-software-properties software-properties-common
4 upgraded, 0 newly installed, 0 to remove and 226 not upgraded.
Need to get 173 kB/203 kB of archives.
After this operation, 46.1 kB disk space will be freed.
Do you want to continue? [Y/n] y
Get:1 http://azure.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apt-transport-https amd64 1.2.32ubuntu0.1 [26.7 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu xenial-updates/main amd64 ca-certificates all 20190110-16.04.1 [146 kB]
Fetched 173 kB in 0s (12.7 MB/s)
Preconfiguring packages ...
(Reading database ... 199478 files and directories currently installed.)
```

```
Processing triggers for systemd (229-4ubuntu21.22) ...
root@vasanthakmindtr:~# docker --version
Docker version 19.03.12, build 48a66213fe
root@vasanthakmindtr:~#
```

```
vasanthakmindtr@vasanthakmindtr:~$ sudo su -
root@vasanthakmindtr:~# git --version
git version 2.7.4
root@vasanthakmindtr:~# apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.7.4-0ubuntu1.9).
0 upgraded, 0 newly installed, 0 to remove and 226 not upgraded.
root@vasanthakmindtr:~# git config --global user.email "kvasanthabecs@gmail.com"
root@vasanthakmindtr:~# git config --global user.username "vasanthkrishnappa"
root@vasanthakmindtr:~#
```



```
Link encap:Ethernet HWaddr 00:0d:3a:a5:bd:bd
inet addr:10.0.1.4 Bcast:10.0.1.255 Mask:255.255.255.0
inet6 addr: fe80::20d:3aff:fea5:bdbd/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:233952 errors:0 dropped:0 overruns:0 frame:0
TX packets:129210 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:206525961 (206.5 MB) TX bytes:46009766 (46.0 MB)

Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:333138 errors:0 dropped:0 overruns:0 frame:0
TX packets:333138 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:178494659 (178.4 MB) TX bytes:178494659 (178.4 MB)

root@vasanthakmindtr:~/jenkin_files#
You have new mail in /var/mail/root
root@vasanthakmindtr:~/jenkin_files#
root@vasanthakmindtr:~/jenkin_files# cat /var/lib/jenkins/secrets/initialAdminPassword
b7b852c89fa448e59505a469287a2f50
root@vasanthakmindtr:~/jenkin_files#
```

The image shows two screenshots of a web browser displaying the Docker Hub interface.

Create Repository Page:

- The URL in the address bar is `hub.docker.com/repository/create`.
- The title is "Create Repository".
- A dropdown menu shows "vasanth95" selected.
- The repository name is "dockerizing_jenkins_pipeline".
- A text input field contains "Dockerizing Jenkins Pipeline project".
- A "Pro tip" box contains the command:

```
docker tag local-image:tagname new-repo:tagname  
docker push new-repo:tagname
```
- A "Visibility" section shows "Public" selected (radio button is checked). The description is "Public repositories appear in Docker Hub search results".
- A "Build Settings (optional)" section notes "Autobuild triggers a new build with every git push to your source code repository." with a link to "Learn More".
- A "Please re-link a GitHub or Bitbucket account" section with a note: "We've updated how Docker Hub connects to GitHub and Bitbucket. You'll need to re-link a GitHub or Bitbucket account." and a search bar.

Repository Details Page:

- The URL in the address bar is `hub.docker.com/repository/docker/vasanth95/dockerizing_jenkins_pipeline`.
- The title is "Repos / vasant95 / dockerizing_jenkins_pipeline".
- The repository name is "vasanth95 / dockerizing_jenkins_pipeline".
- A "General" tab is selected, showing "Using 0 of 1 private repositories." with a "Get more" link.
- A "Docker commands" section contains the command:

```
docker push  
vasanth95/dockerizing_jenkins_pipeline:tagname
```
- A "Tags" section notes "This repository is empty. When it's not empty, you'll see a list of the most recent tags here."
- A "Recent builds" section notes "Link a source provider and run a build to see build results here."
- A "Readme" section notes "Repository description is empty. Click [here](#) to edit."

A screenshot of a Microsoft Edge browser window. The address bar shows the URL github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline. The page displays information about a repository named "Dockerizing-Jenkins-Pipeline". It includes sections for "Code", "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings". A prominent section titled "Quick setup — if you've done this kind of thing before" provides instructions for setting up the repository, including a command-line script:

```
echo "# Dockerizing-Jenkins-Pipeline" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
git push -u origin master
```

A screenshot of a Microsoft Edge browser window. The address bar shows the URL 10.0.1.4:8080/view/all/newjob. The page is titled "Jenkins" and shows a form for creating a new item. The input field contains "Docker jenkins intergration pipeline project" with the note "» Required field". Below the input field, there is a section titled "Freestyle project" with a description: "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." A "Pipeline" button is visible at the bottom of this section. The browser's taskbar at the bottom shows icons for File, Home, Task View, Start, Taskbar settings, and Search.

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Pipeline Syntax" under the "Docker jenkins intergration pipeline project". The page displays a configuration form for a Jenkins pipeline step. On the left, there's a sidebar with links like "Declarative Directive Generator", "Declarative Online Documentation", "Steps Reference", "Global Variables Reference", "Online Documentation", "Examples Reference", and "IntelliJ IDEA GDSL". The main form has a "Sample Step" dropdown set to "git: Git". Below it, the "Repository URL" is set to "https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git", "Branch" is set to "master", and "Credentials" are set to "vasanthkrishnappa/******** (git user creds)". There are two checked checkboxes at the bottom: "Include in polling?" and "Include in changelog?".



The screenshot shows a "Add Credentials" dialog box overlaid on a Jenkins pipeline configuration page. The dialog has fields for "Domain" (set to "Global credentials (unrestricted)"), "Kind" (set to "Username with password"), "Scope" (set to "Global (Jenkins, nodes, items, all child items, etc)"), "Username" (set to "vasanthkrishnappa"), "Password" (set to "*****"), and "ID" (set to "gituser"). The background Jenkins page is visible, showing the same pipeline configuration as the first screenshot.

The screenshot shows two stacked windows from a web browser.

The top window displays the Jenkins Pipeline Syntax Snippet Generator. It has a "Generate Pipeline Script" button at the top. Below it is a code editor containing the following Groovy script:

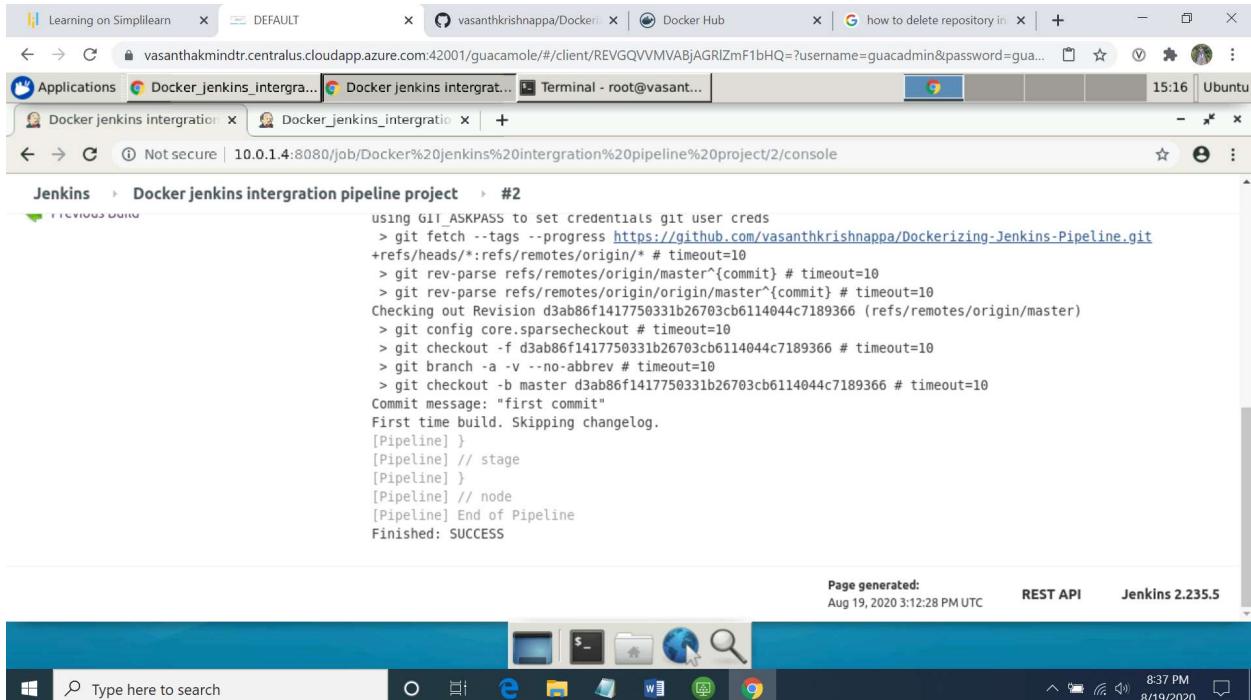
```
git credentialsId: 'gituser', url: 'https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git'
```

Below the code editor is a section titled "Global Variables" with the following text:

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the [Global Variables Reference](#) for details.

At the bottom right, it says "Page generated: Aug 19, 2020 2:56:40 PM UTC" and "Jenkins 2.235.5".

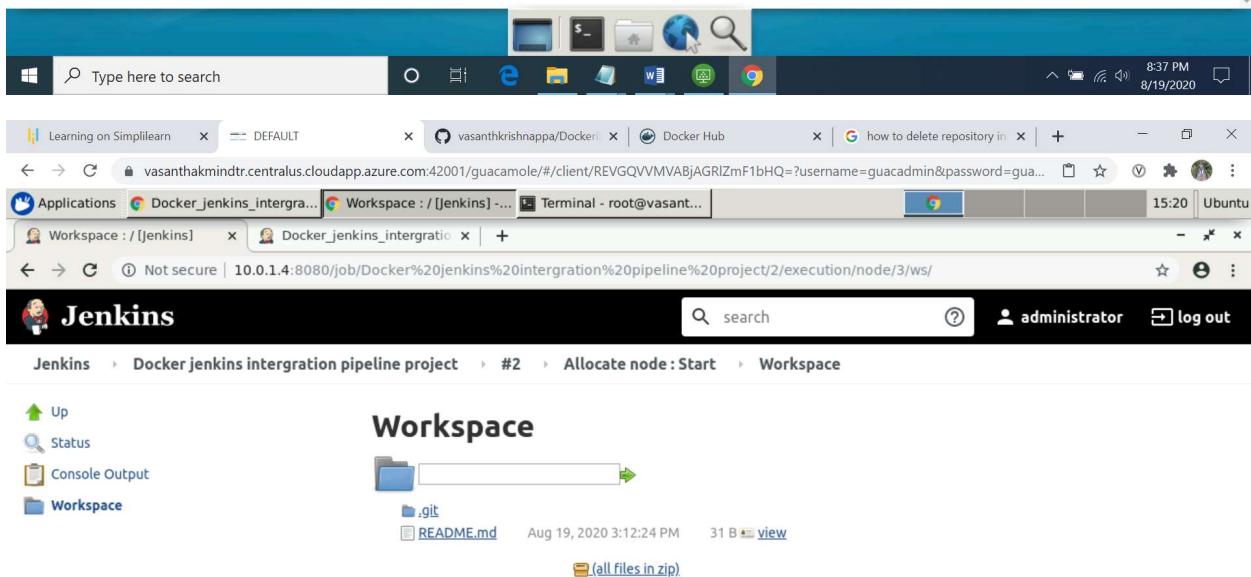
The bottom window shows the Jenkins Pipeline configuration page for a project named "Docker jenkins intergration". The "Pipeline" tab is selected. Under "Definition", the "Pipeline script" tab is active, showing the same Groovy script as above. The "Save" and "Apply" buttons are visible at the bottom of the script editor.



Jenkins > Docker jenkins intergration pipeline project > #2

```
using GIT_ASKPASS to set credentials git user creds
> git fetch --tags --progress https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision d3ab86f1417750331b26703cb6114044c7189366 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f d3ab86f1417750331b26703cb6114044c7189366 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git checkout -b master d3ab86f1417750331b26703cb6114044c7189366 # timeout=10
Commit message: "first commit"
First time build. Skipping changelog.
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Page generated: Aug 19, 2020 3:12:28 PM UTC REST API Jenkins 2.235.5



Jenkins > Docker jenkins intergration pipeline project > #2 > Allocate node : Start > Workspace

Workspace

- Up
- Status
- Console Output
- Workspace**

.git
README.md Aug 19, 2020 3:12:24 PM 31 B view
[\(all files in zip\)](#)

Page generated: Aug 19, 2020 3:20:22 PM UTC Jenkins 2.235.5



The screenshot shows the Webhook Relay management interface. A modal window is open, titled "New token created". It displays two fields: "Key" containing the value "ade26373-0157-4a76-828f-368af7ade934" and "Secret" containing the value "vyG58GCJa3w9". Below these fields, there are instructions for configuring the relay CLI and creating Kubernetes secrets. The background shows the main dashboard with various navigation options like Dashboard, Buckets, Relay logs, Functions, Tunnels, Access tokens, Domain Registrar, Integrations, Connected services, and Webhooks.

The screenshot shows a terminal session on an Ubuntu system. The user is configuring the relay CLI by running the following commands:

```
relay login -k ade26373-0157-4a76-828f-368af7ade934 -s vyG58GCJa3w9
export RELAY_KEY=ade26373-0157-4a76-828f-368af7ade934
export RELAY_SECRET=vyG58GCJa3w9
```

After running these commands, the user exports the relay key and secret as environment variables:

```
export RELAY_KEY=ade26373-0157-4a76-828f-368af7ade934
export RELAY_SECRET=vyG58GCJa3w9
```

The terminal also shows the configuration of a webhook relay agent, including setting up forwarding and starting the webhook relay agent.

The screenshot shows the GitHub Settings interface for a repository. On the left, a sidebar lists various settings categories: Options, Manage access, Security & analysis, Branches, Webhooks (which is currently selected and highlighted in red), Notifications, Integrations, Deploy keys, Secrets, Actions, Moderation, and Interaction limits. The main content area is titled "Webhooks / Add webhook". It contains instructions about sending POST requests to a URL with event details. A "Payload URL" input field contains the value "https://oIwkgx6phxnun8wifpcjej.hooks.webhookrelay.com". Below it, a "Content type" dropdown is set to "application/x-www-form-urlencoded". There is a "Secret" input field which is empty. Under "SSL verification", there is a note that SSL certificates are verified by default. Two radio buttons are present: "Enable SSL verification" (selected) and "Disable (not recommended)". At the bottom, a section asks "Which events would you like to trigger this webhook?", but no specific events are listed.

The screenshot shows the Jenkins Pipeline configuration for a project named "Docker Jenkins Intergration pipeline project". The "Build Triggers" tab is active. It lists several trigger options: "Build after other projects are built", "Build periodically", "GitHub hook trigger for GITScm polling" (which is checked), "Poll SCM", "Disable this project", "Quiet period", and "Trigger builds remotely (e.g., from scripts)". Each option has a help icon (a question mark inside a blue circle) to its right. At the bottom of the list are two buttons: "Save" and "Apply". The background shows the Jenkins dashboard with other projects and a terminal window labeled "[Terminal - root@vasan...]".

```
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git add README.md
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>" to unstage)

    modified:   README.md

root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git commit -m "2 commit"
[master 0186b1d] 2 commit
 1 file changed, 1 insertion(+)
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git push -u origin master
Username for 'https://github.com': vasanthkrishnappa
Password for 'https://vasanthkrishnappa@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 292 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
  d3ab86f..0186b1d master -> master
Branch master set up to track remote branch master from origin.
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline#
```

Jenkins > Docker jenkins intergration pipeline project > #3

Previous Build

```
> git fetch --tags --progress https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 0186b1d7cce633f393d34f6d39f70ea9550bba58 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 0186b1d7cce633f393d34f6d39f70ea9550bba58 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D master # timeout=10
> git checkout -b master 0186b1d7cce633f393d34f6d39f70ea9550bba58 # timeout=10
Commit message: "2 commit"
> git rev-list --no-walk d3ab86f1417750331b26703cb6114044c7189366 # timeout=10
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Page generated:
Aug 19, 2020 3:53:17 PM UTC REST API Jenkins 2.235.5

```
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git add README.md
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>" to unstage)

    modified:   README.md

root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git commit -m "2 commit"
[master 0186b1d] 2 commit
 1 file changed, 1 insertion(+)
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git push -u origin master
Username for 'https://github.com': vasanthkrishnappa
Password for 'https://vasanthkrishnappa@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 292 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
  d3ab86f..0186b1d master -> master
Branch master set up to track remote branch master from origin.
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline#
```

```
Jenkins > Docker Jenkins Intergration Pipeline project > #3
  Previous Build
> git fetch --tags --progress https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking Out Revision 0186b1d7cce633f393d34f6d39f70ea9550bba58 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 0186b1d7cce633f393d34f6d39f70ea9550bba58 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D master # timeout=10
> git checkout -b master 0186b1d7cce633f393d34f6d39f70ea9550bba58 # timeout=10
Commit message: "2 commit"
> git rev-list --no-walk d3ab86f1417750331b26703cb6114044c7189366 # timeout=10
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Page generated:
Aug 19, 2020 3:53:17 PM UTC REST API Jenkins 2.235.5

A screenshot of a Windows desktop environment. At the top, there is a taskbar with various application icons. Below the taskbar, a browser window is open to a Jenkins project named "Docker Jenkins intergration". The browser tabs show "Applications" and "Terminal - root@vasanth...". The main content of the browser is a terminal window titled "Terminal - root@vasanthakmindtr: ~/Dockerizing-Jenkins-Pipeline". The terminal session shows the following command history:

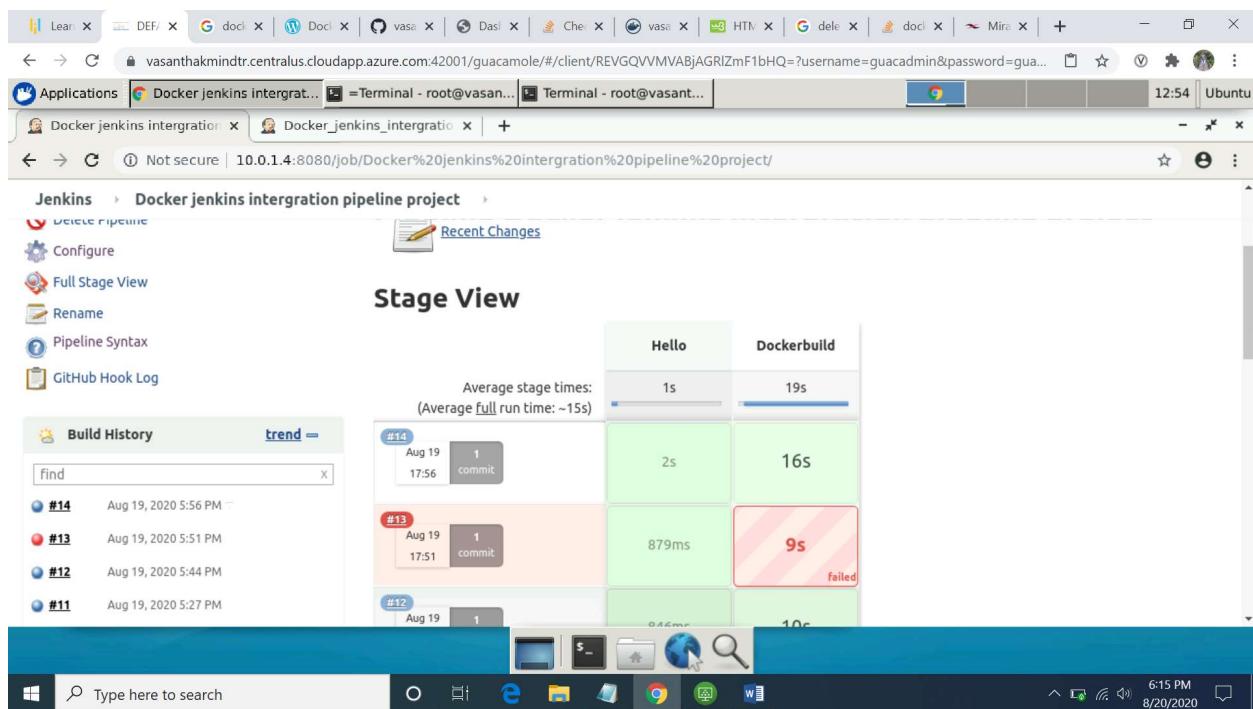
```
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# vi login.html
No protocol specified
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git add login.html
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git add Dockerfile
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:  login.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    nginx.conf

root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git commit -m "committing change"
[master a6fdcad] committing change
 1 file changed, 5 insertions(+), 3 deletions(-)
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git push -u origin master
Username for 'https://github.com': vasanthkrishnappa
Password for 'https://vasanthkrishnappa@github.com':
```



```
modified: login.html  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    nginx.conf  
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git commit -m "commiting change"  
[master a6fdcad] committing change  
 1 file changed, 5 insertions(+), 3 deletions(-)  
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git push -u origin master  
Username for 'https://github.com': vasanthkrishnappa  
Password for 'https://vasanthkrishnappa@github.com':  
Counting objects: 3, done.  
Delta compression using up to 2 threads.  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 406 bytes | 0 bytes/s, done.  
Total 3 (delta 1), reused 0 (delta 0)  
remote: Resolving deltas: 100% (1/1), completed with 1 local object.  
To https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git  
 * [new branch] master -> master  
Branch master set up to track remote branch master from origin.  
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline#
```

```
Hit:9 http://ppa.launchpad.net/certbot/certbot/ubuntu xenial InRelease  
Hit:10 http://ppa.launchpad.net/linuxuprising/java/ubuntu xenial InRelease  
Hit:11 http://ppa.launchpad.net/remmina-ppa-team/freerdp-daily/ubuntu xenial InRelease  
Hit:12 http://ppa.launchpad.net/ubuntu-mozilla-security/ppa/ubuntu xenial InRelease  
Get:13 https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages [14.5 kB]  
Hit:14 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease  
Fetched 299 kB in 1s (186 kB/s)  
Reading package lists... Done  
root@vasanthakmindtr:~# apt-get install docker-ce docker-ce-cli containerd.io  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed:  
  containerd.io docker-ce docker-ce-cli  
0 upgraded, 3 newly installed, 0 to remove and 226 not upgraded.  
Need to get 84.4 MB of archives.  
After this operation, 381 MB of additional disk space will be used.  
Get:1 https://download.docker.com/linux/ubuntu xenial/stable amd64 containerd.io amd64 1.2.13-2 [20.9 MB]  
Get:2 https://download.docker.com/linux/ubuntu xenial/stable amd64 docker-ce-cli amd64 5:19.03.12-3-0-ubuntu-xenial [41.0 MB]  
Get:3 https://download.docker.com/linux/ubuntu xenial/stable amd64 docker-ce amd64 5:19.03.12-3-0-ubuntu-xenial [22.4 MB]  
Fetched 84.4 MB in 1s (42.7 MB/s)  
Selecting previously unselected package containerd.io.
```

Screenshot of a web browser showing the Jenkins Stage View for a Docker Jenkins integration pipeline project. The pipeline consists of two stages: 'Hello' and 'Dockerbuild'. The 'Hello' stage has an average time of 1s. The 'Dockerbuild' stage has an average time of 19s. The build history shows the following builds:

Build	Date	Commit	Time
#15	Aug 20, 2020 12:55 PM	No Changes	1s
#14	Aug 19, 2020 5:56 PM	1 commit	12s
#13	Aug 19, 2020 5:51 PM		2s
#12	Aug 19, 2020 5:44 PM		16s
#11			

The Jenkins interface includes a 'Recent Changes' link and various configuration options like 'Delete Pipeline', 'Configure', 'Full Stage View', 'Rename', 'Pipeline Syntax', and 'GitHub Hook Log'.

Screenshot of a web browser showing the Jenkins Stage View for the same Docker Jenkins integration pipeline project at a later time. The build history now includes builds #16 through #7. Builds #15 and #13 failed, while others succeeded.

Build	Date	Commit	Time	Status	
#16	(pending—In the quiet period. Expires in 1.7 sec)		3s	Pending	
#15	Aug 20, 2020 12:55 PM	1 commit	13s	failed	
#14	Aug 19, 2020 5:56 PM	1 commit	2s	Passed	
#13	Aug 19, 2020 5:51 PM	1 commit	879ms	9s	failed
#12	Aug 19, 2020 5:44 PM	1 commit	846ms	10s	Passed
#11	Aug 19, 2020 5:27 PM	1 commit	1s	13s	Passed
#10	Aug 19, 2020 5:16 PM				
#9	Aug 19, 2020 5:12 PM				
#8	Aug 19, 2020 4:57 PM				
#7	Aug 19, 2020 4:54 PM				

Jenkins > Docker jenkins intergration pipeline project > #18

```
83t428/e1f04: Waiting
7ef368776582: Waiting
8682t9a74649: Layer already exists
b10158c8df3a: Layer already exists
95317c6d9303: Layer already exists
7e5d31cfbd2c: Layer already exists
83t4287e1f04: Layer already exists
d3a6da143c91: Layer already exists
7ef368776582: Layer already exists
c2efc452f941: Pushed
webserver: digest: sha256:edd954321aa7fb88c76834aab631f7f406e054bcd9604545648b51e665ded size: 1990
+ docker run -d -p 80:81 --name=httpserver vasanth95/dockerizing_jenkins_pipeline:webserver
d038bcfe733b0a0467d846d669fc86aaab53b4380226b68e1bedfeaafble2f78
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

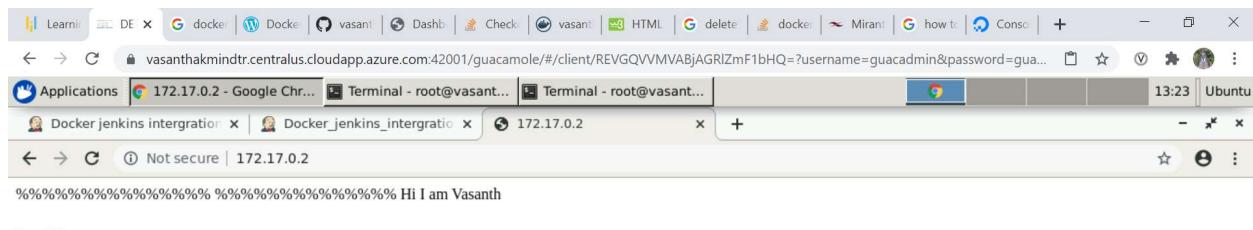
Page generated: Aug 20, 2020 1:18:57 PM UTC REST API Jenkins 2.235.5

Type here to search

Terminal - root@vasanthakmindtr: ~/Dockerizing-Jenkins-Pipeline

```
File Edit View Terminal Tabs Help
udp6      0      0 fe80::20d:3aff:fea5:123 :::*
2234/ntpd
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
STATUS              PORTS NAMES
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# vi login.html
No protocol specified
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git add login.html
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git commit -m "commit"
[master 2933f01] commit
 1 file changed, 1 insertion(+), 1 deletion(-)
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git push -u origin master
Username for 'https://github.com': vasanthkrishnappa
Password for 'https://vasanthkrishnappa@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 280 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
 9a06eba..2933f01  master -> master
Branch master set up to track remote branch master from origin.
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline#
```

Page generated: Aug 20, 2020 1:18:57 PM UTC REST API Jenkins 2.235.5



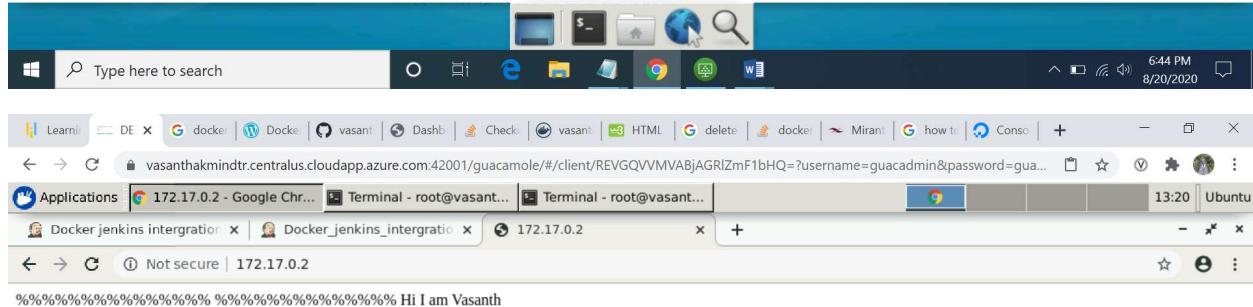
%%%%% Hi I am Vasanth

hello

this is simply learn project

This is a paragraph.

This is another paragraph.



%%%%% Hi I am Vasanth

hello

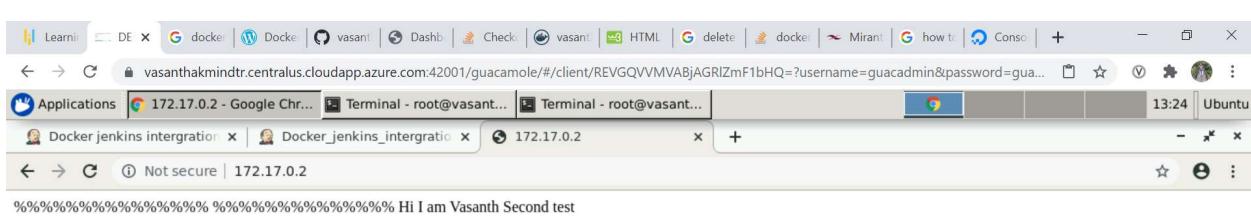
this is simply learn project

This is a paragraph.

This is another paragraph.



```
CONTAINER ID        IMAGE               COMMAND      CREATED             STATUS              PORTS               NAMES
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# vi login.html
No protocol specified
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git add login.html
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git commit -m "commit"
[master d437c3f] commit
 1 file changed, 1 insertion(+), 1 deletion(-)
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git push -u origin master
Username for 'https://github.com': Vasanth#95
Password for 'https://Vasanth#95@github.com':
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline# git push -u origin master
Username for 'https://github.com': vasanthkrishnappa
Password for 'https://vasanthkrishnappa@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 285 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/vasanthkrishnappa/Dockerizing-Jenkins-Pipeline.git
 2933f01..d437c3f  master -> master
Branch master set up to track remote branch master from origin.
root@vasanthakmindtr:~/Dockerizing-Jenkins-Pipeline#
```



%%%%%%%%%%%%%%% Hi I am Vasantha Second test

hello

this is simply learn project

This is a paragraph.

This is another paragraph.



New Pricing and Packaging announced! [Learn more...](#)

Explore Repositories Organizations Get Help vasant95

Repositories vasant95 / dockerizing_jenkins_pipeline Using 0 of 1 private repositories. [Get more](#)

General Tags Builds Timeline Collaborators Webhooks Settings

Action Filter Tags Sort by Latest

IMAGE webserver Last updated in 4 minutes by vasant95

DIGEST 9d1d93d69278 OS/ARCH linux/amd64

COMRESSED SIZE 76.19 MB

docker pull vasant95/dockerizing_jenkiin

The screenshot shows a GitHub repository page for 'Dockerizing-Jenkins-Pipeline'. The repository was created by 'vasanthkrishnappa' and has 18 commits. The README.md file contains the following content:

```
Dockerizing-Jenkins-Pipeline
the testing the auto trigger
```