DevOps Certification training -End Project

Build a Docker Jenkins Pipeline to Implement CI/CD Workflow

Project Agenda:
To build a Docker Jenkins Pipeline to Implement CI/CD Workflow

Description:

Create simple DevOps project to show how using DevOps tools I am going to build Docker Jenkins pipeline

Tools required:

GitHub – Git – Jenkins – Docker - Docker Hub

Git , Jenkins and Docker All of these tools must installed in the machine what we work on it.

And must have create an accounts on GitHub, Docker Hub and Jenkins

Expected Deliverables:
Create a GitHub repository, clone the GitHub repository, create project files using Git and push them to GitHub repository, build and push project image with docker, and build and push image to Docker Hub ,create pipeline job in Jenkins .

Developed By:
KHOLOOOD IBRAHEM

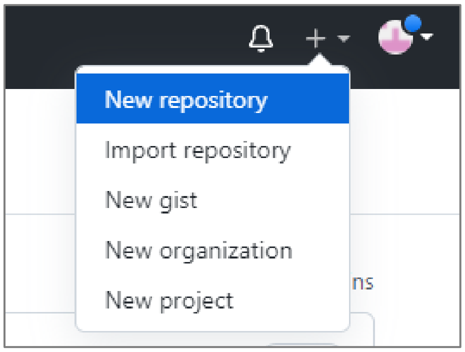
Steps to be followed:

1. Create a new repository and Clone the GitHub repository
2. Add project's files to local repository
3. Push the project's files to the remote repository in GitHub
4. Build Docker mage for project
5. Push image to Docker Hub
6. Build pipeline job with Jenkins

Step 1: Creating a new repository and Clone the GitHub repository

1.1 Open the browser in your lab, go to **github.com,** and log in to my account (my account username is *“kholoodi11”*)

1.2 Then click on the **+** icon from the upper-right corner of the page and select **New repository** from the drop-down menu

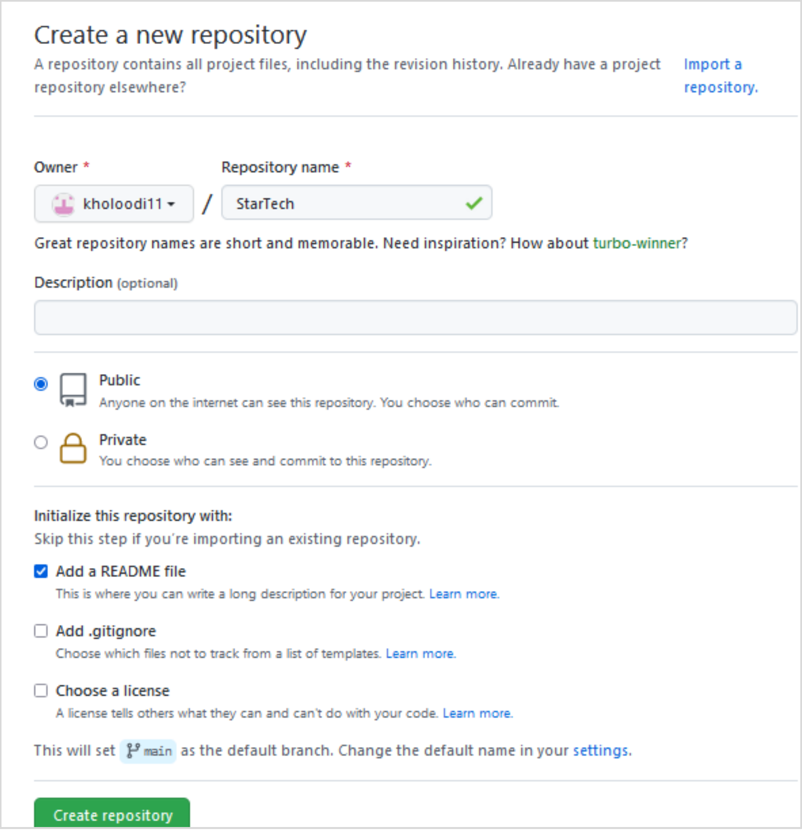


1.3 Now, enter *StarTech* **Repository name.** Entering **Description** is optional.

1.4 Choose **Public** for the repository type

1.5 Select **Initialize this repository with a README** to include a *README* file for the repository

1.6 Click on the **Create Repository** button



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * / Repository name *

kholoodi11 / StarTech ✓

Great repository names are short and memorable. Need inspiration? How about turbo-winner?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☑ Add a README file
This is where you can write a long description for your project. Learn more.

☐ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

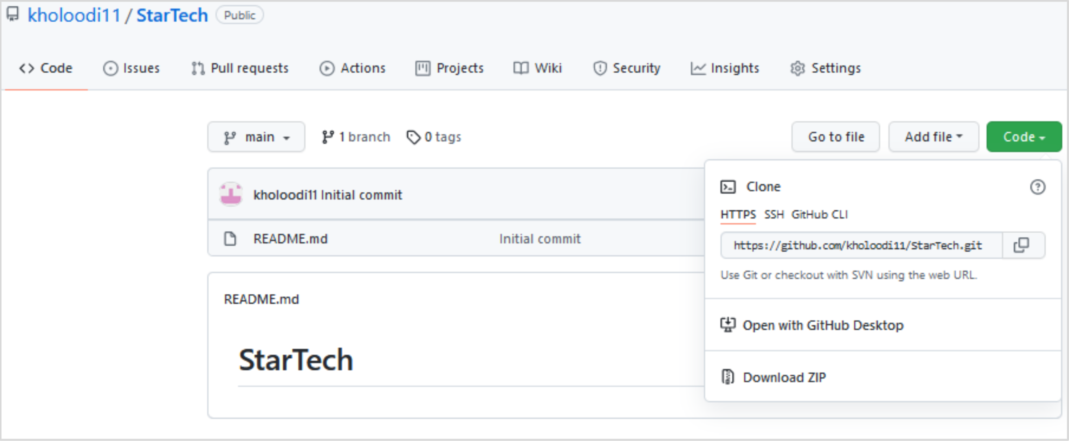
☐ Choose a license
A license tells others what they can and can't do with your code. Learn more.

This will set main as the default branch. Change the default name in your settings.

Create repository

1.7 Then open the **StarTech** repository and click on the **Code** button

1.8 And click on the icon as shown in the following image to copy the **URL** provided under **HTTPS ,**copy the URL

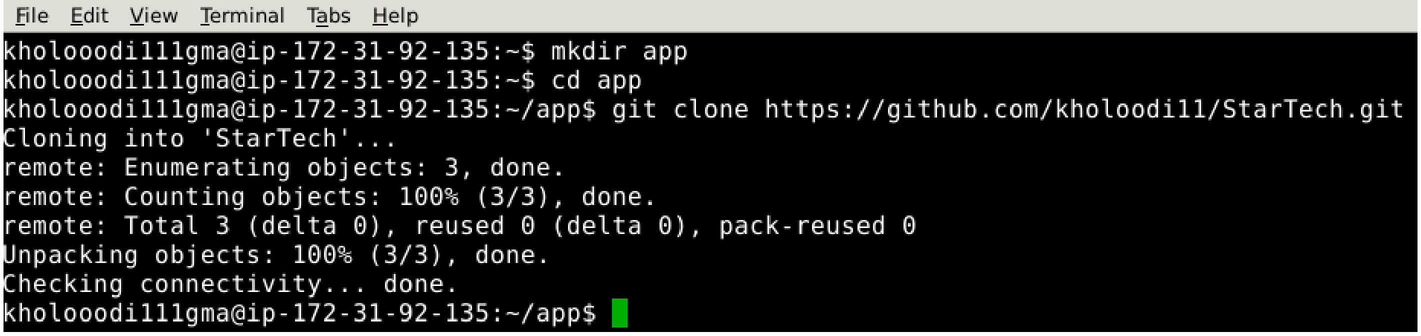


```
https://github.com/kholoodi11/StarTech.git
```

1.9 Open the **terminal** and create a new directory and in this directory clone the **StarTech** repository from GitHub use the following command:

$ git clone<the URL of repo that copied in 1.8)

The change current directory to **app by** command ***cd:***



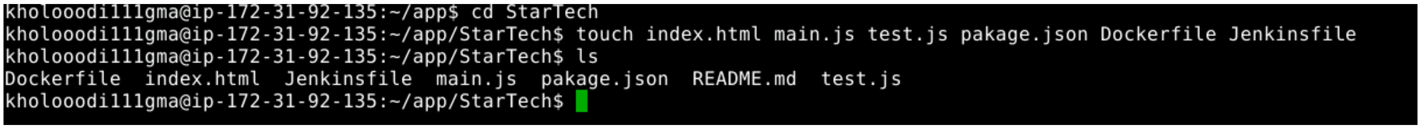
```
kholooodi111gma@ip-172-31-92-135:~$ mkdir app
kholooodi111gma@ip-172-31-92-135:~$ cd app
kholooodi111gma@ip-172-31-92-135:~/app$ git clone https://github.com/kholoodi11/StarTech.git
Cloning into 'StarTech'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
kholooodi111gma@ip-172-31-92-135:~/app$
```

Now I have local repository for **star_tech** project

Step 2: Add project's files to local repository

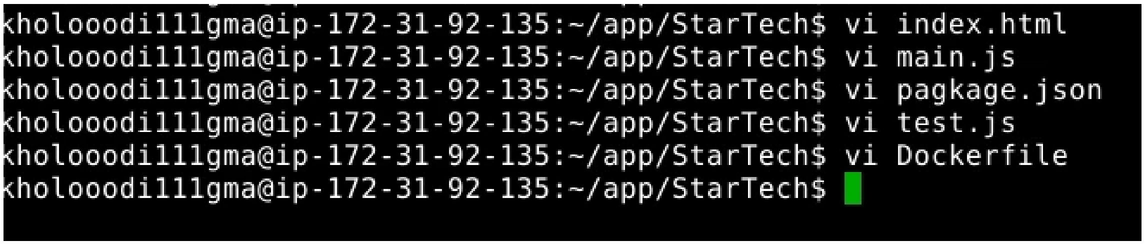
2.1 By using Linux and Git command create and add project files to locally **StarTech**

2.2 Navigate the StarTech repository , create files via **touch** command and list the files **via ls**



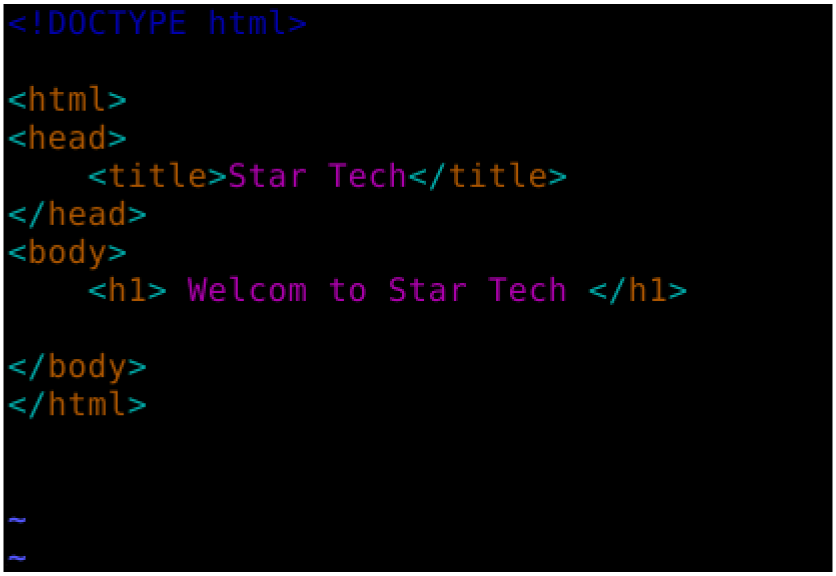
```
kholooodi111gma@ip-172-31-92-135:~/app$ cd StarTech
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ touch index.html main.js test.js pakage.json Dockerfile Jenkinsfile
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ ls
Dockerfile  index.html  Jenkinsfile  main.js  pakage.json  README.md  test.js
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$
```

2.3 Add content to files using vi command with file name



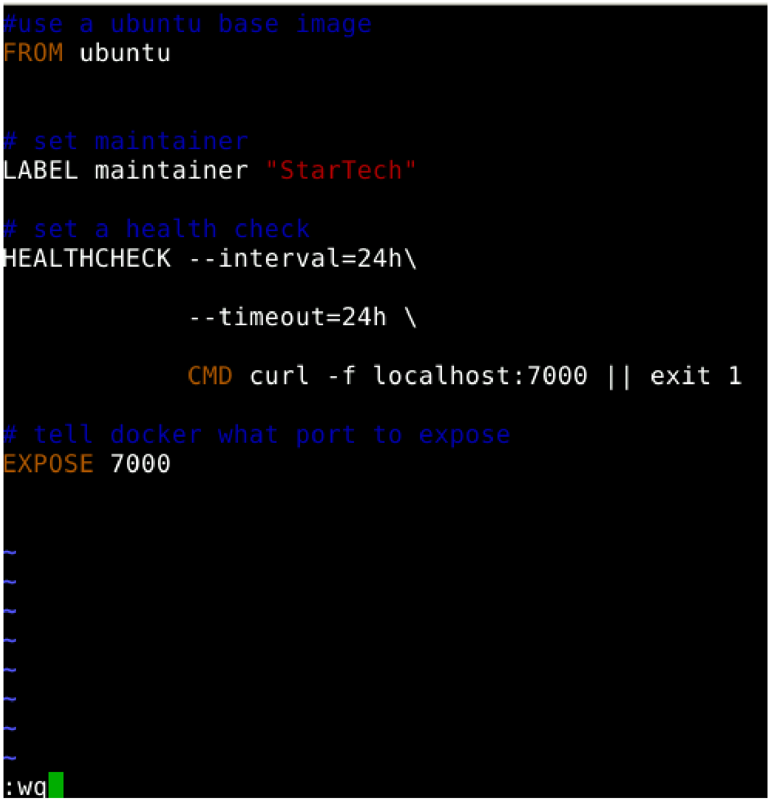
```
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ vi index.html
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ vi main.js
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ vi pagkage.json
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ vi test.js
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ vi Dockerfile
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$
```

Index.html



```
<!DOCTYPE html>

<html>
<head>
    <title>Star Tech</title>
</head>
<body>
    <h1> Welcom to Star Tech </h1>

</body>
</html>



~
~
```

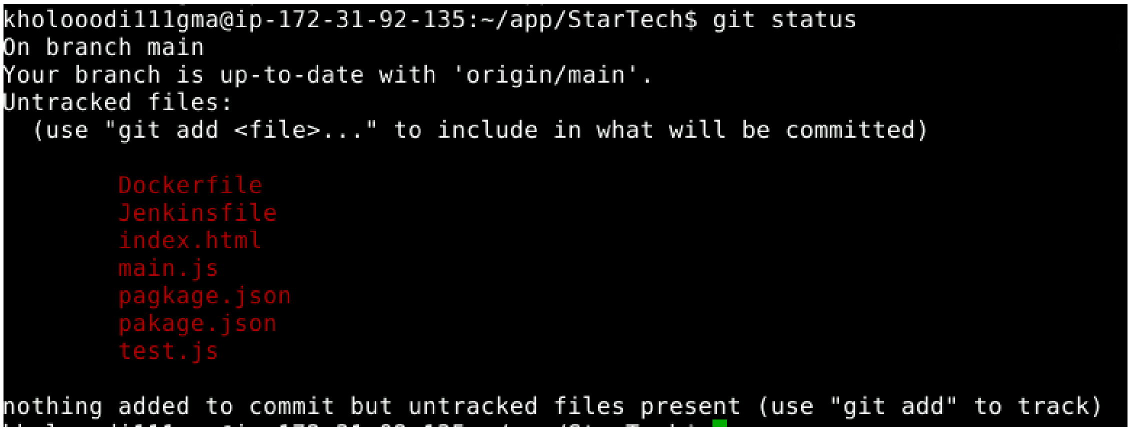
Docker file



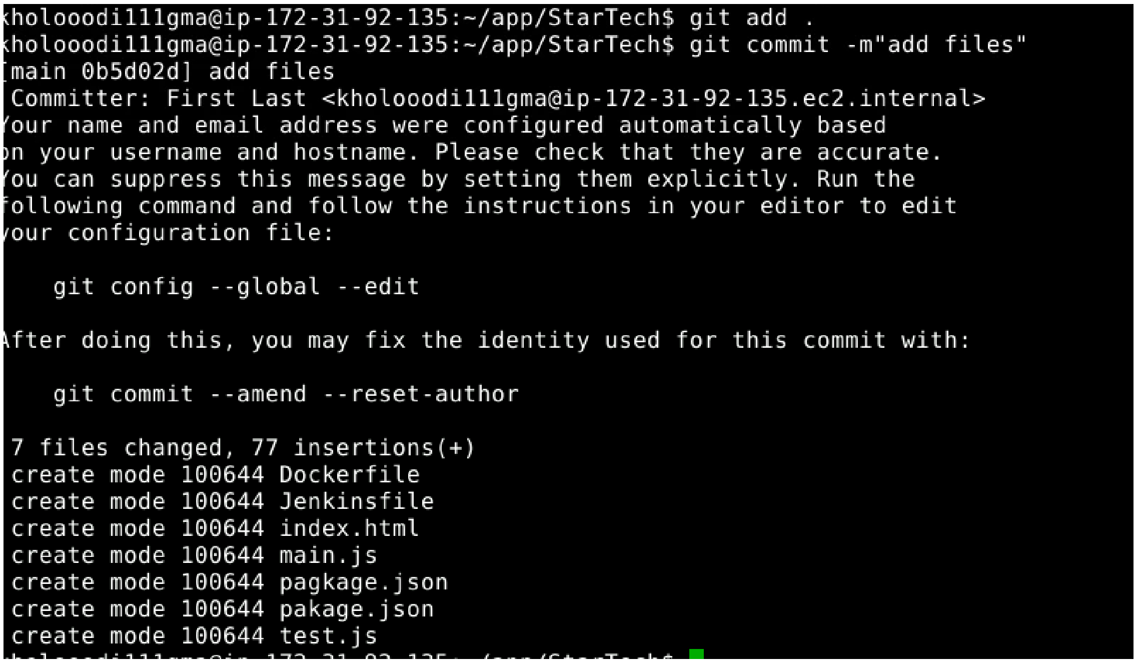
```
#use a ubuntu base image
FROM ubuntu


# set maintainer
LABEL maintainer "StarTech"

# set a health check
HEALTHCHECK --interval=24h\

            --timeout=24h \

            CMD curl -f localhost:7000 || exit 1

# tell docker what port to expose
EXPOSE 7000



~
~
~
~
~
~
~
~
:wq
```

2.4 Now by using Git command add the files to local repository

Git status , git add , git commit -m “commit message ”

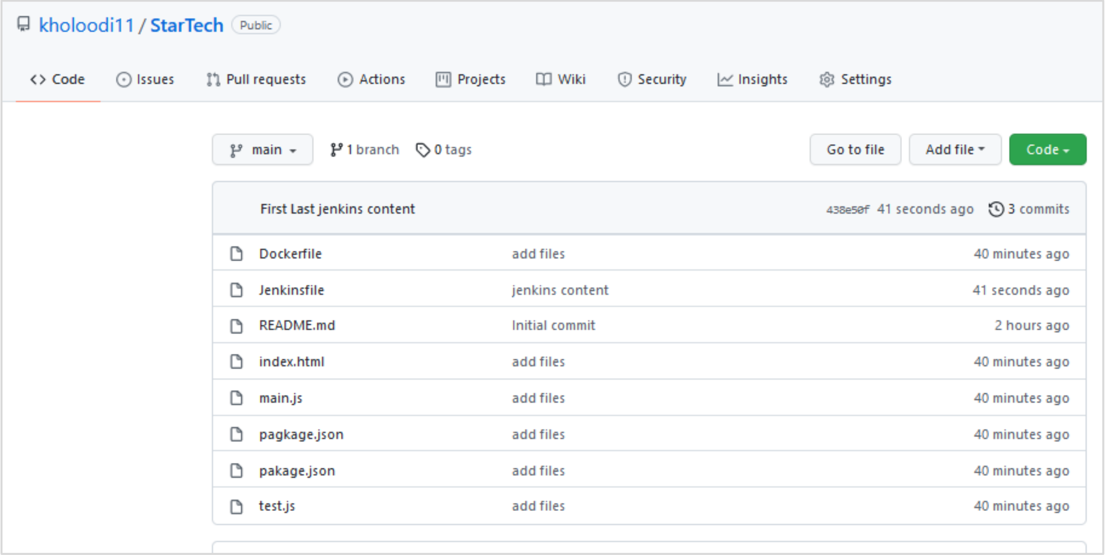


```
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ git status
On branch main
Your branch is up-to-date with 'origin/main'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Dockerfile
        Jenkinsfile
        index.html
        main.js
        pagkage.json
        pakage.json
        test.js

nothing added to commit but untracked files present (use "git add" to track)
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$
```



```
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ git add .
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ git commit -m"add files"
[main 0b5d02d] add files
 Committer: First Last <kholooodi111gma@ip-172-31-92-135.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 7 files changed, 77 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 Jenkinsfile
 create mode 100644 index.html
 create mode 100644 main.js
 create mode 100644 pagkage.json
 create mode 100644 pakage.json
 create mode 100644 test.js
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$
```

2. 5 Then push the files to GitHub repository with git push command

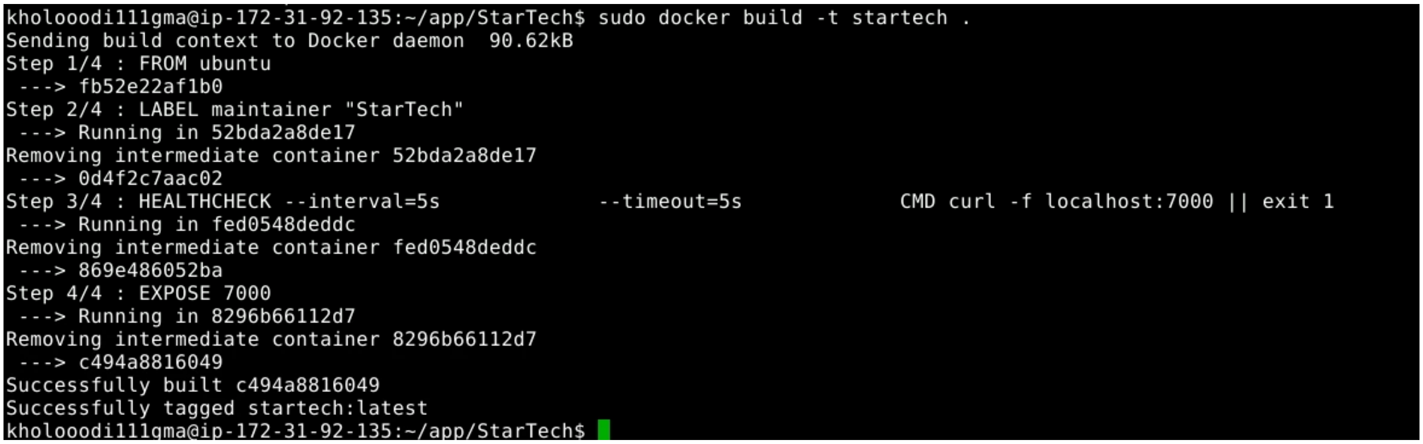


First Last jenkins content 438e50f 41 seconds ago 3 commits

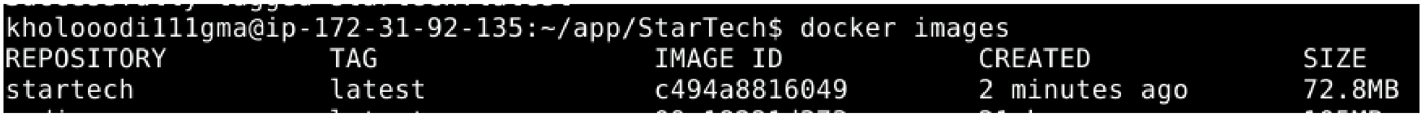
Dockerfile	add files	40 minutes ago
Jenkinsfile	jenkins content	41 seconds ago
README.md	Initial commit	2 hours ago
index.html	add files	40 minutes ago
main.js	add files	40 minutes ago
pagkage.json	add files	40 minutes ago
pakage.json	add files	40 minutes ago
test.js	add files	40 minutes ago

Step 3: Build image for project

3.1 Build the image via docker command **docker build**



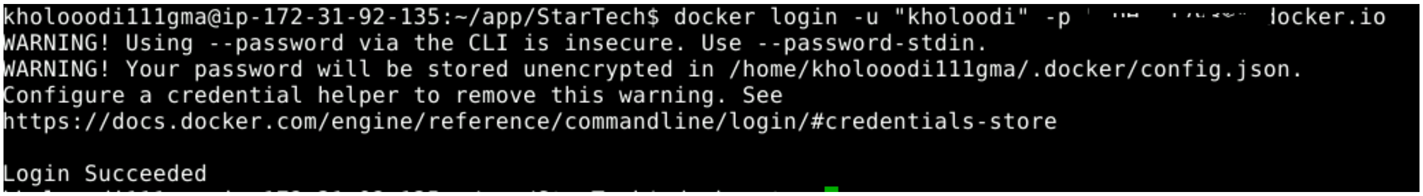
```
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ sudo docker build -t startech .
Sending build context to Docker daemon  90.62kB
Step 1/4 : FROM ubuntu
 ---> fb52e22af1b0
Step 2/4 : LABEL maintainer "StarTech"
 ---> Running in 52bda2a8de17
Removing intermediate container 52bda2a8de17
 ---> 0d4f2c7aac02
Step 3/4 : HEALTHCHECK --interval=5s             --timeout=5s             CMD curl -f localhost:7000 || exit 1
 ---> Running in fed0548deddc
Removing intermediate container fed0548deddc
 ---> 869e486052ba
Step 4/4 : EXPOSE 7000
 ---> Running in 8296b66112d7
Removing intermediate container 8296b66112d7
 ---> c494a8816049
Successfully built c494a8816049
Successfully tagged startech:latest
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$
```



```
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ docker images
REPOSITORY           TAG                 IMAGE ID            CREATED             SIZE
startech             latest              c494a8816049        2 minutes ago       72.8MB
```

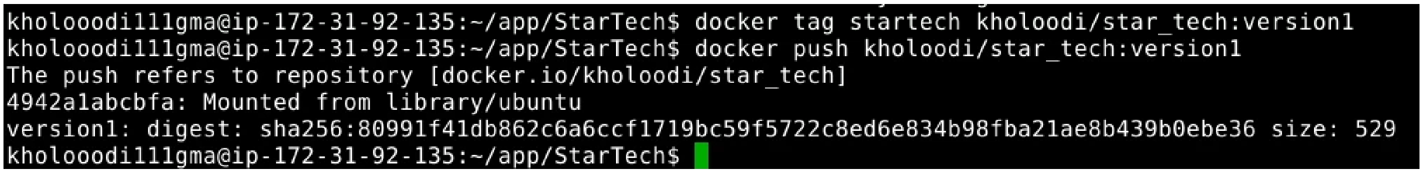
Step 4: push image to Docker Hub

4.1 login to docker hub account



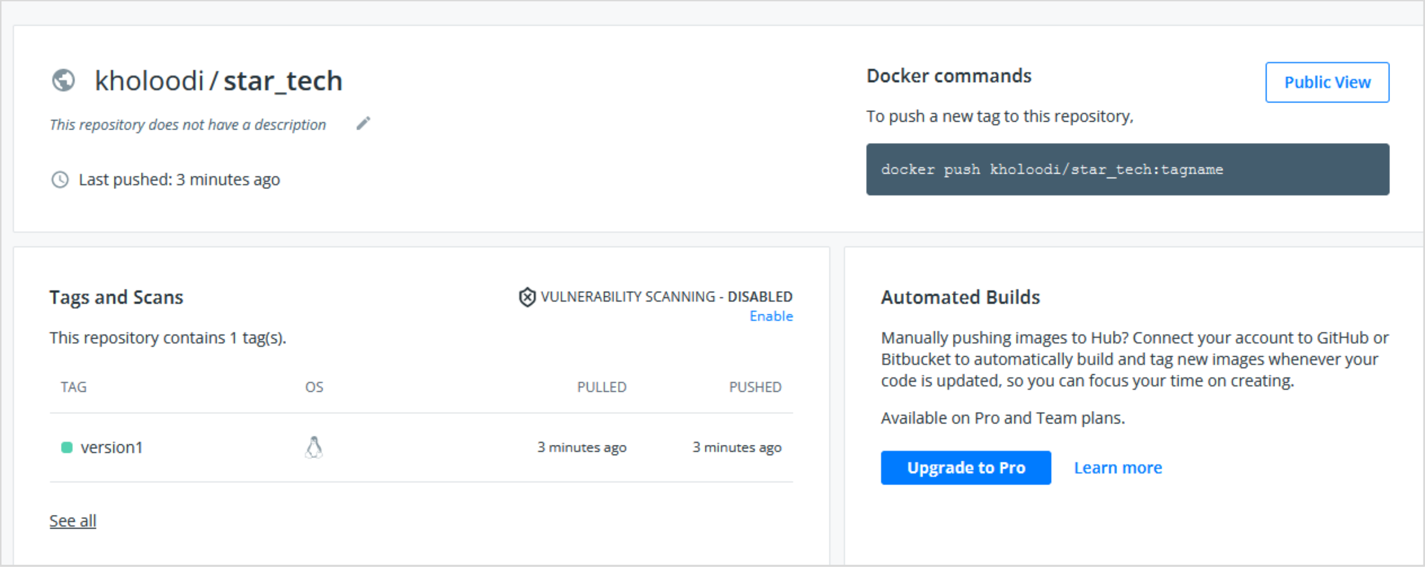
```
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ docker login -u "kholoodi" -p [redacted] docker.io
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /home/kholooodi111gma/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

4.2 push the image



```
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ docker tag startech kholoodi/star_tech:version1
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$ docker push kholoodi/star_tech:version1
The push refers to repository [docker.io/kholoodi/star_tech]
4942a1abcbfa: Mounted from library/ubuntu
version1: digest: sha256:80991f41db862c6a6ccf1719bc59f5722c8ed6e834b98fba21ae8b439b0ebe36 size: 529
kholooodi111gma@ip-172-31-92-135:~/app/StarTech$
```

The image pushed to docker hub



kholoodi / star_tech

This repository does not have a description

Last pushed: 3 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push kholoodi/star_tech:tagname
```

Tags and Scans

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
version1		3 minutes ago	3 minutes ago

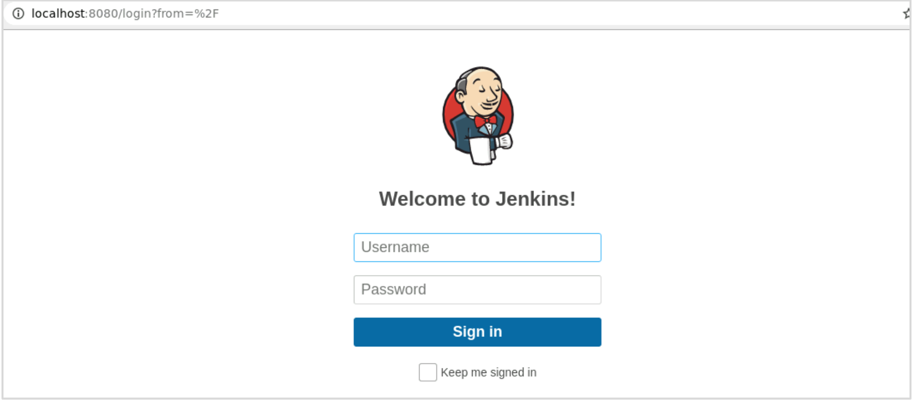
Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

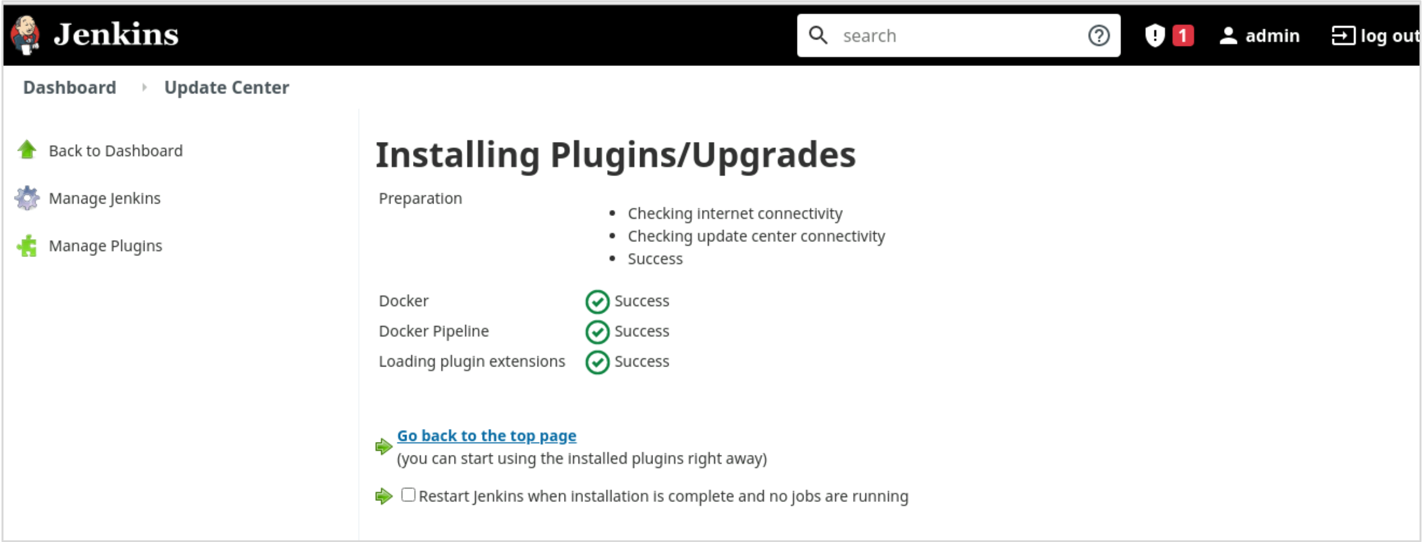
Available on Pro and Team plans.

Step 5: Build Pipeline job with Jenkins

5.1 Login to Jenkins account



5.2 Before doing Pipeline with docker we need install some docker plugins like docker & docker pipeline



Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Docker ✓ Success

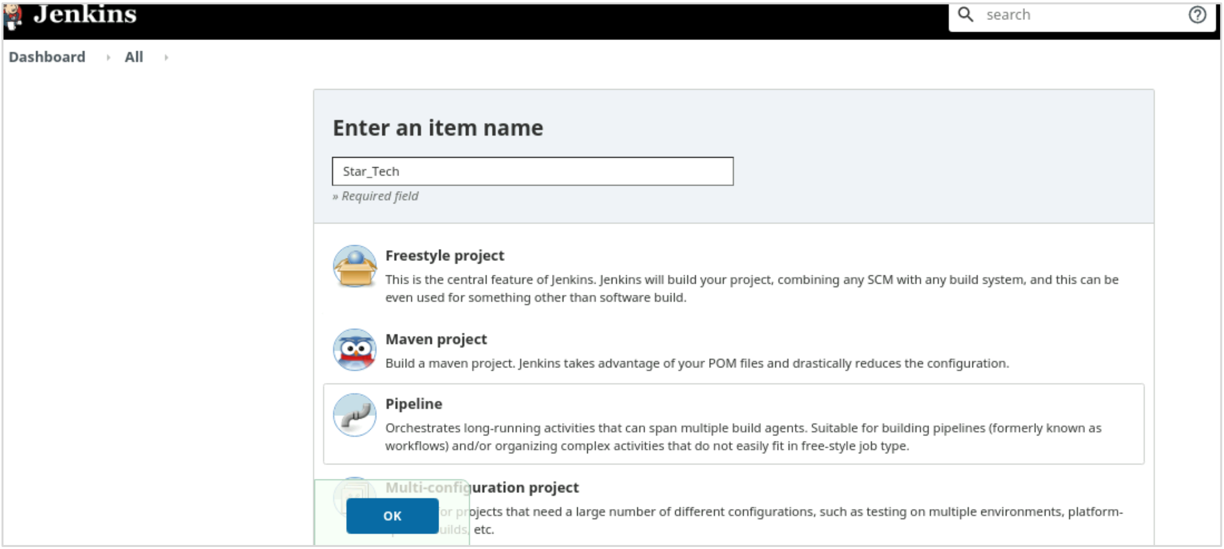
Docker Pipeline ✓ Success

Loading 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

5.3 In the dashboard options choose **New item** to create new job choose **Pipeline** as type name of this job then **Ok**



Jenkins

search

Dashboard › All ›

Enter an item name

Star_Tech

» 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.

Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline

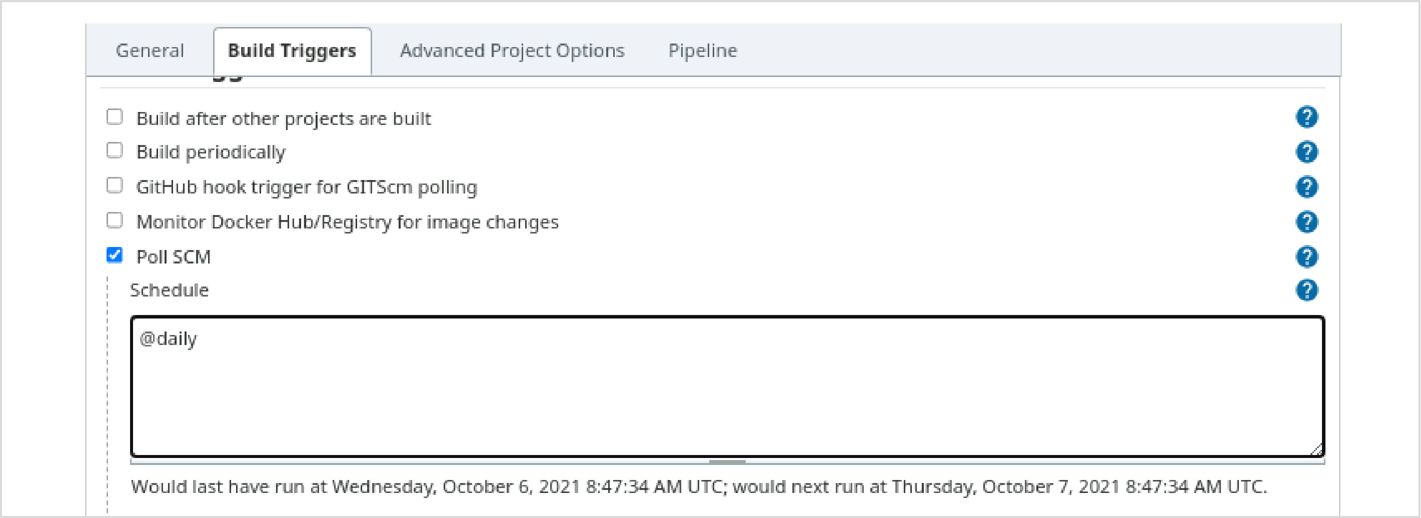
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project

...or projects that need a large number of different configurations, such as testing on multiple environments, platform-...uilds, etc.

OK

5.4 And for Build Triggers choose Poll SCM everyday



General | **Build Triggers** | Advanced Project Options | Pipeline

□ Build after other projects are built

□ Build periodically

□ GitHub hook trigger for GITScm polling

□ Monitor Docker Hub/Registry for image changes

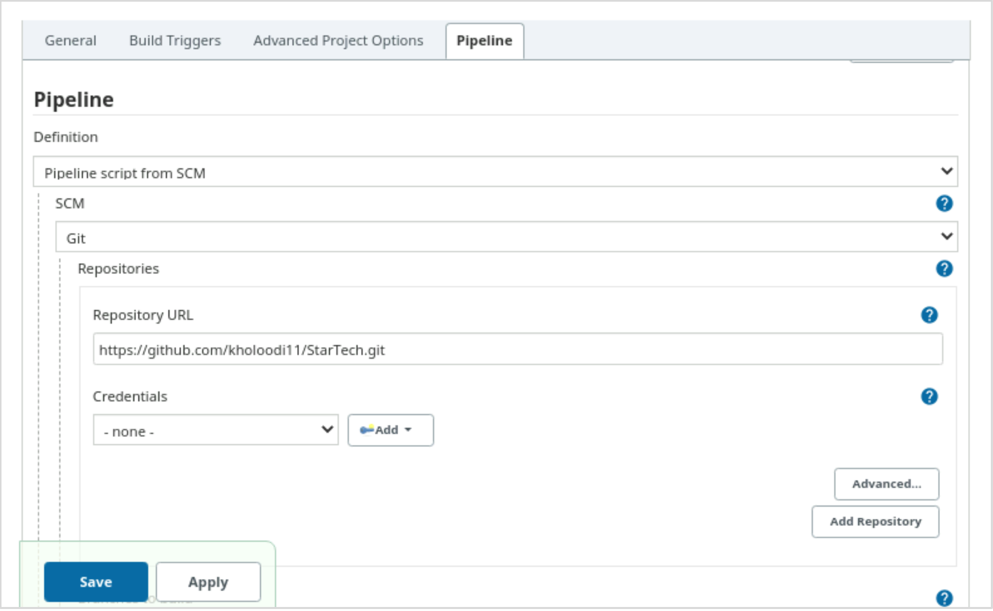
☑ Poll SCM

Schedule

```
@daily
```

Would last have run at Wednesday, October 6, 2021 8:47:34 AM UTC; would next run at Thursday, October 7, 2021 8:47:34 AM UTC.

5.5 Now, for definition choose **Pipeline scrip from SCM** add SCM choose **Git and** the URL of GitHub repository for **Repository URL**



Pipeline

Definition: Pipeline script from SCM

SCM: Git

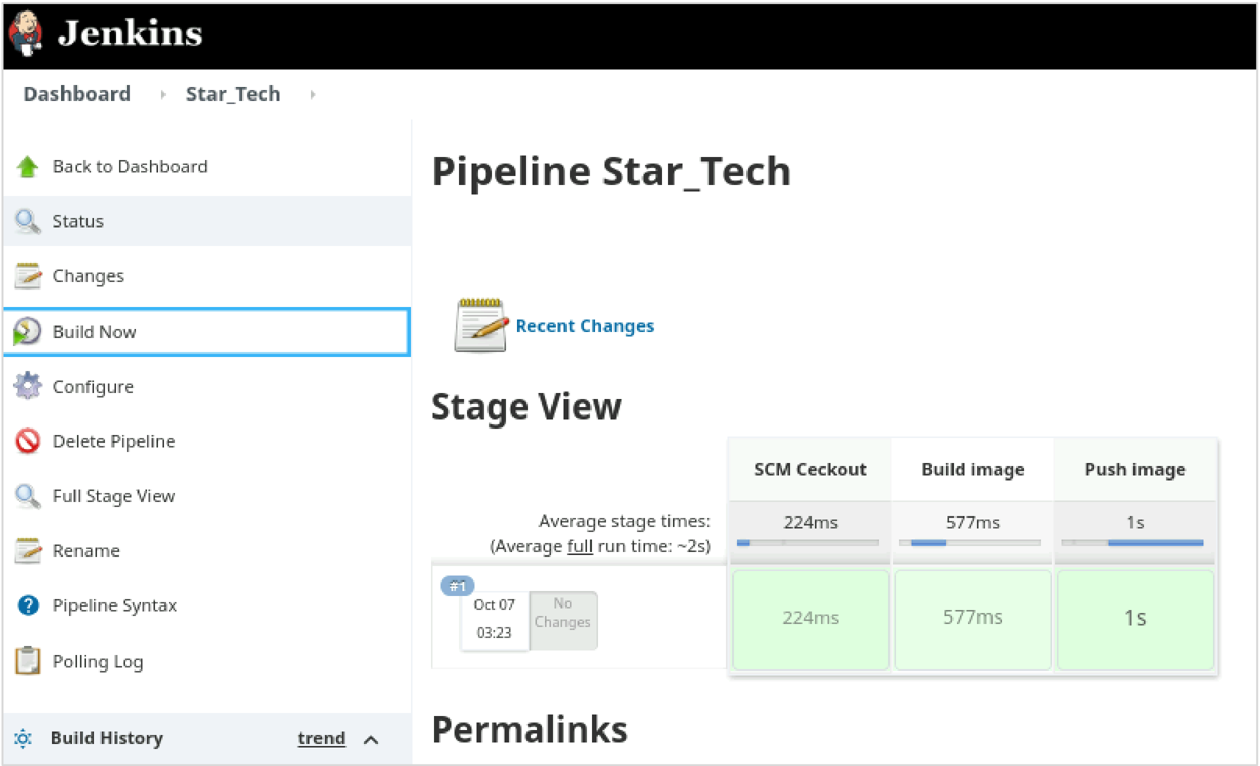
Repositories

Repository URL: https://github.com/kholoodi11/StarTech.git

Credentials: - none -

Then Apply and save

5.6 For Build the job click on **Build Now**



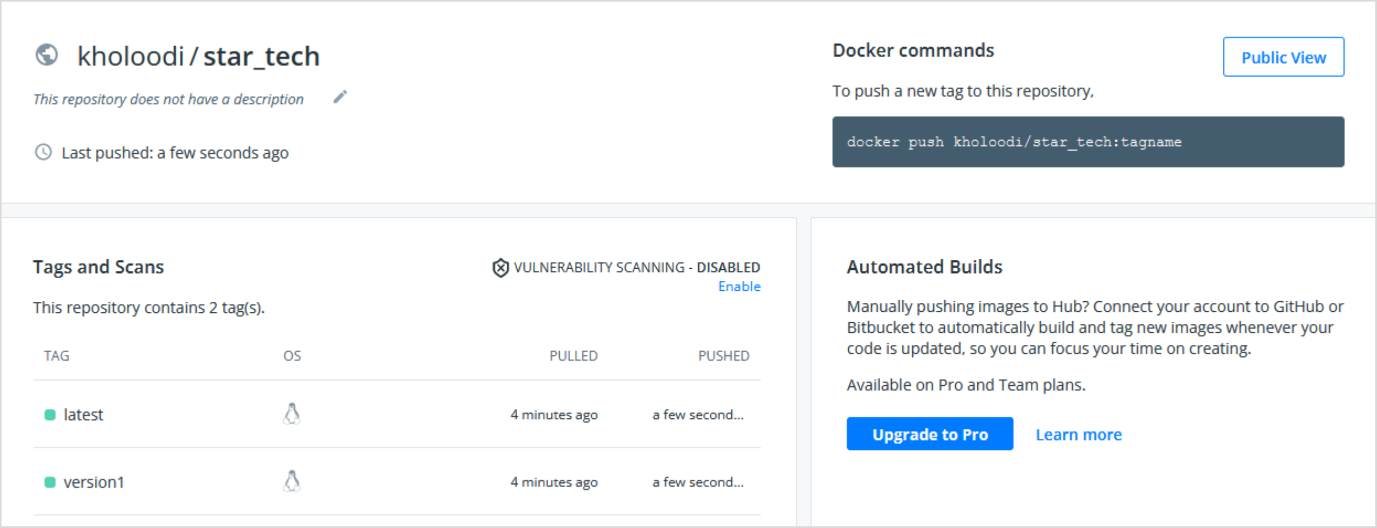
Pipeline Star_Tech

Stage View

	SCM Ceckout	Build image	Push image
Average stage times: (Average full run time: ~2s)	224ms	577ms	1s
#1 Oct 07 03:23 No Changes	224ms	577ms	1s

Permalinks

See the update version pushed to Docker Hub



kholoodi / star_tech

This repository does not have a description

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository,

```
docker push kholoodi/star_tech:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED

This repository contains 2 tag(s).

TAG	OS	PULLED	PUSHED
latest	🐧	4 minutes ago	a few second...
version1	🐧	4 minutes ago	a few second...

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available on Pro and Team plans.