Jenkins Task_1

1- What is Jenkins used for?

2- Install jenkins with docker image.

3- what is plugins in jenkins ?

4- Create free style project and link it to private git repo containing any dockerfile then build an image from this dockerfile and push it to private docker repo

The Solutions :

1 - What is Jenkins used for?

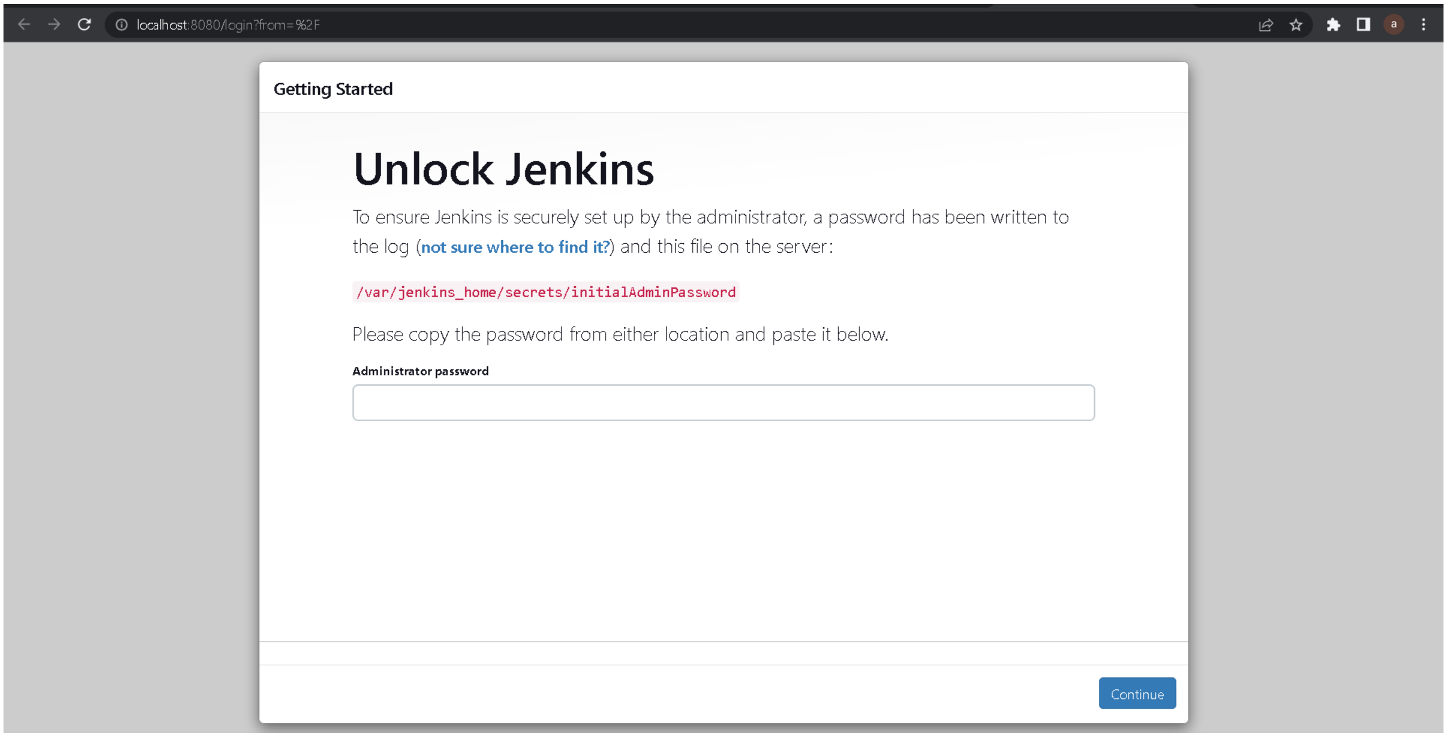
Jenkins is an open-source automation server that is widely used for continuous integration and continuous delivery (CI/CD) processes. It helps automate the building, testing, and deployment of software applications

2 - Install jenkins with docker image.

1 - docker pull jenkins/Jenkins

2 - docker run -d -p 8080:8080 -p 50000:50000 jenkins/Jenkins

3 – docker ps to check



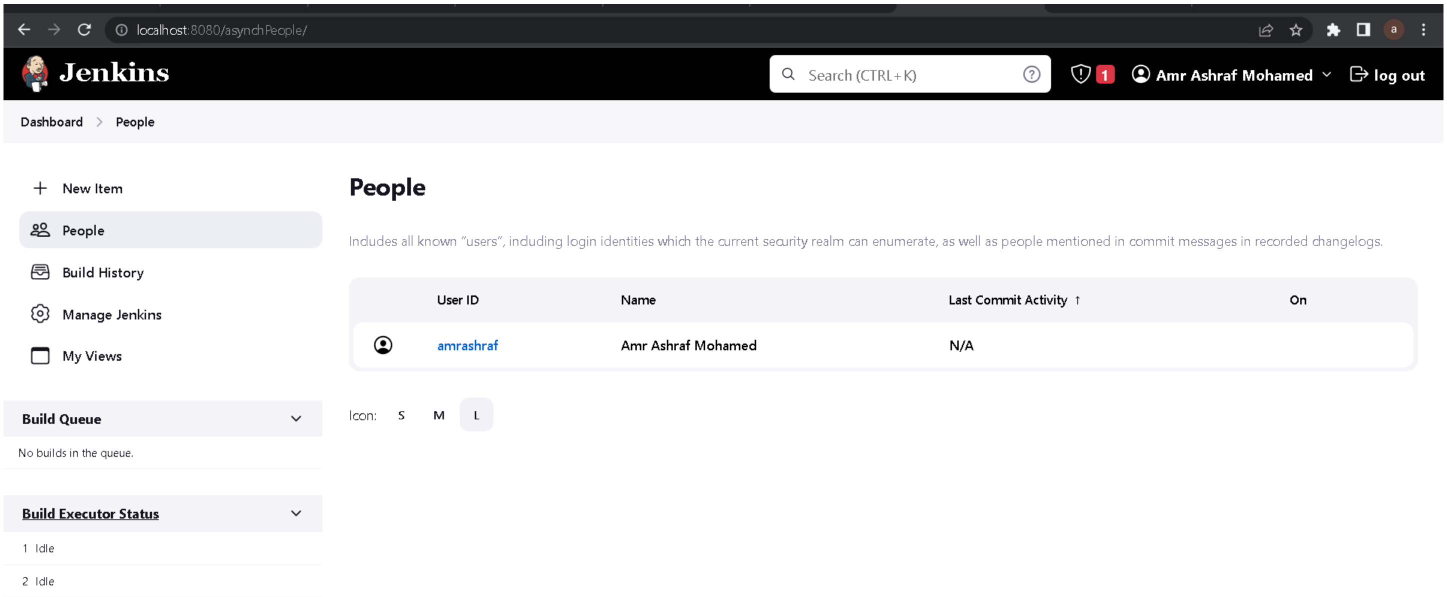
Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password



People

Includes all known "users", including login identities which the current security realm can enumerate, as well as people mentioned in commit messages in recorded changelogs.

	User ID	Name	Last Commit Activity ↑	On
	amrashraf	Amr Ashraf Mohamed	N/A	

No builds in the queue.

1 Idle

2 Idle

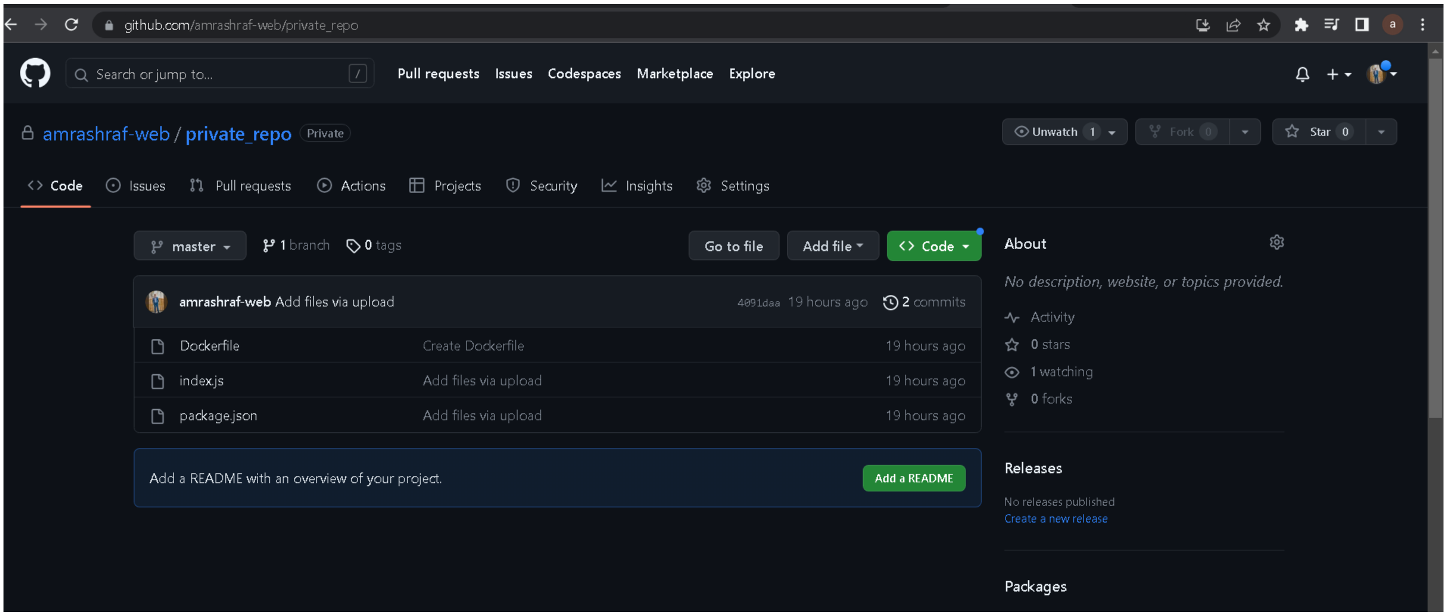
3 - what is plugins in jenkins ?

Plugins in Jenkins are extensions that provide additional functionality and features to the Jenkins server. Jenkins has a vast plugin ecosystem that allows you to add and configure plugins to tailor the automation server according to your specific needs. Plugins can provide integrations with various tools, enable new types of projects, extend reporting capabilities, and much more.

4 - Create free style project and link it to private git repo containing any dockerfile then build an image from this dockerfile and push it to private docker repo

1 – create private repository on my github

Link : https://github.com/amrashraf-web/private_repo



amrashraf-web Add files via upload 4091daa 19 hours ago 2 commits

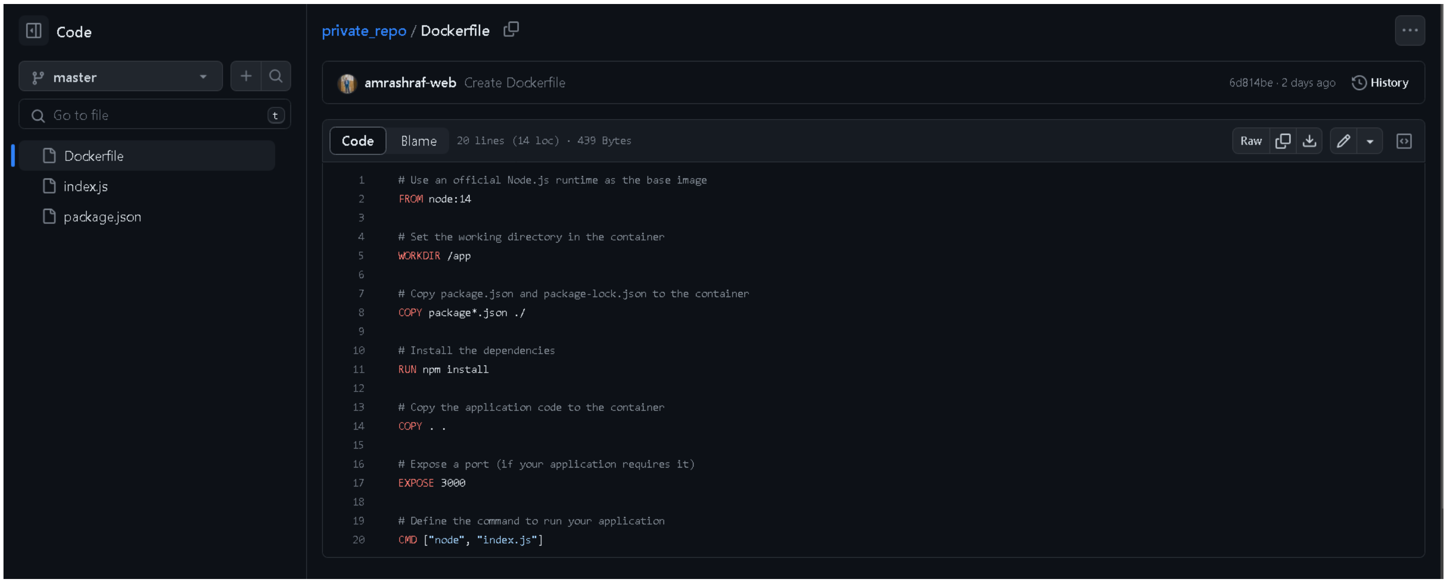
Dockerfile	Create Dockerfile	19 hours ago
index.js	Add files via upload	19 hours ago
package.json	Add files via upload	19 hours ago

Add a README with an overview of your project.

About

No description, website, or topics provided.

No releases published

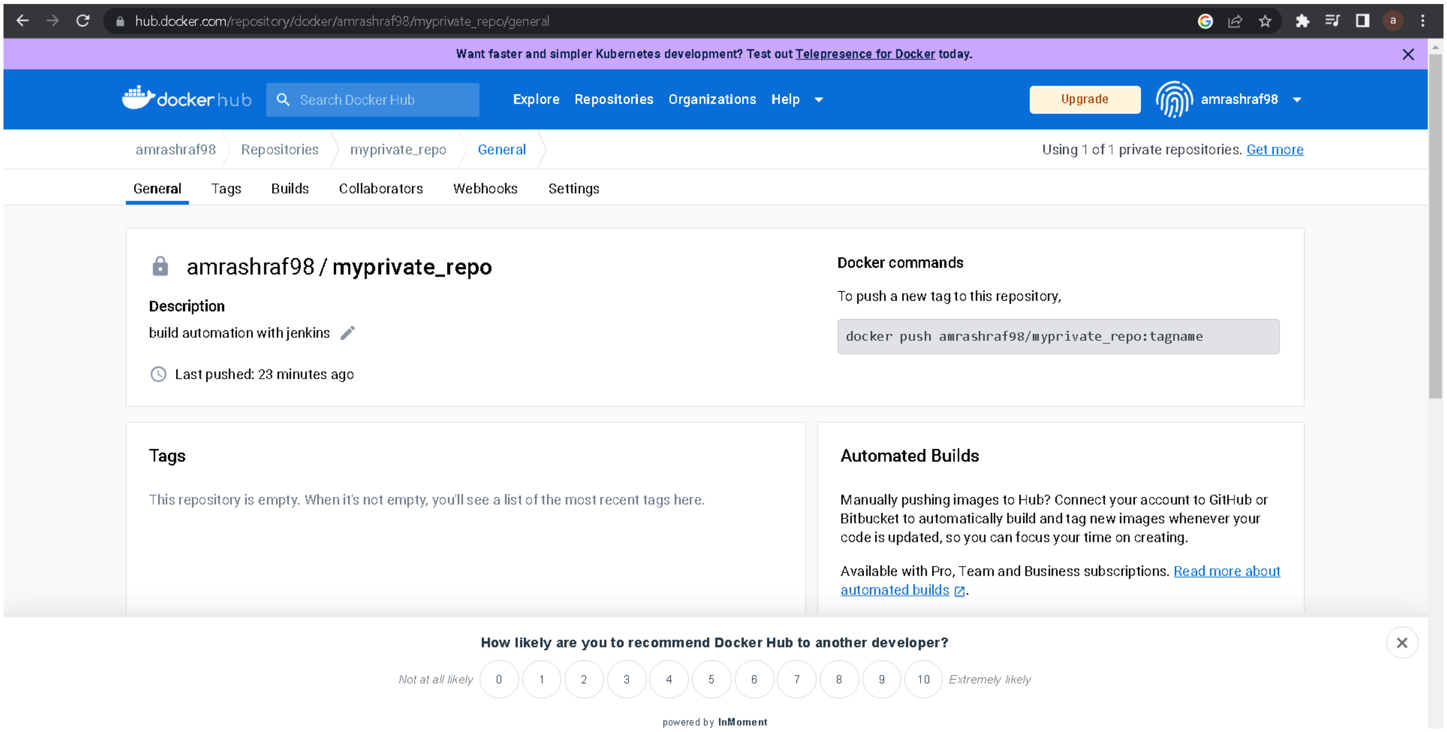


amrashraf-web Create Dockerfile 6d814be · 2 days ago

20 lines (14 loc) · 439 Bytes

```
# Use an official Node.js runtime as the base image
FROM node:14

# Set the working directory in the container
WORKDIR /app

# Copy package.json and package-lock.json to the container
COPY package*.json ./

# Install the dependencies
RUN npm install

# Copy the application code to the container
COPY . .

# Expose a port (if your application requires it)
EXPOSE 3000

# Define the command to run your application
CMD ["node", "index.js"]
```

2 – create private repo on docker



Want faster and simpler Kubernetes development? Test out Telepresence for Docker today.

Using 1 of 1 private repositories. Get more

amrashraf98 / **myprivate_repo**

Description

build automation with jenkins

Last pushed: 23 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push amrashraf98/myprivate_repo:tagname
```

Tags

This repository is empty. When it's not empty, you'll see a list of the most recent tags here.

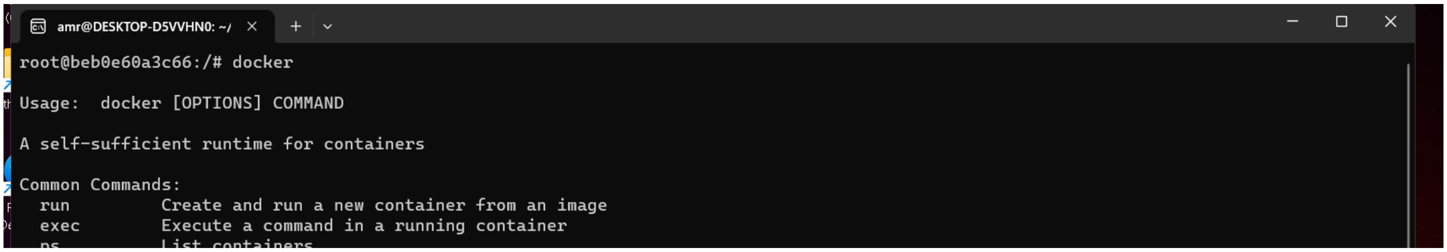
Automated Builds

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 with Pro, Team and Business subscriptions. Read more about automated builds.

How likely are you to recommend Docker Hub to another developer?

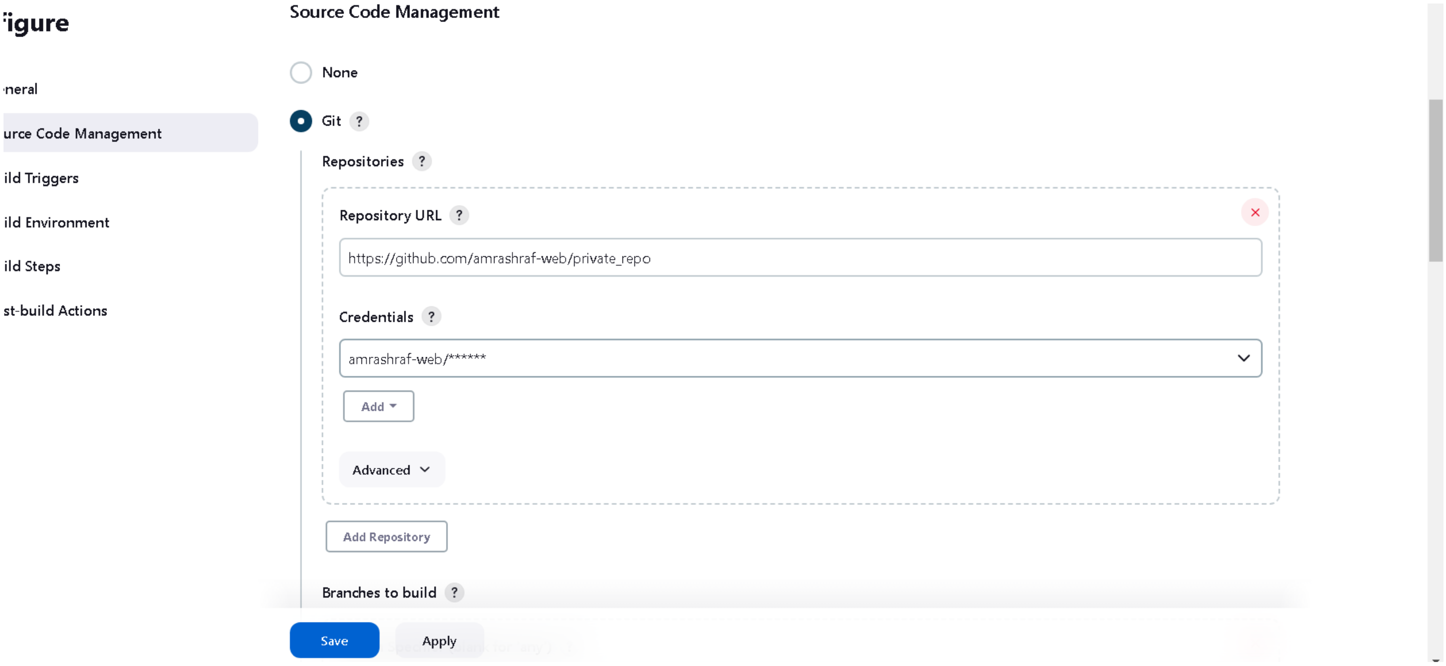
3 – install docker on Jenkins server to build my Docker file



```
root@beb0e60a3c66:/# docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
  run         Create and run a new container from an image
  exec        Execute a command in a running container
  ps          List containers
```

4 - create free style project on Jenkins with credentials user and token of GitHub + commands for docker build and push with image

For configuration project



Source Code Management

○ None

● Git

Repositories

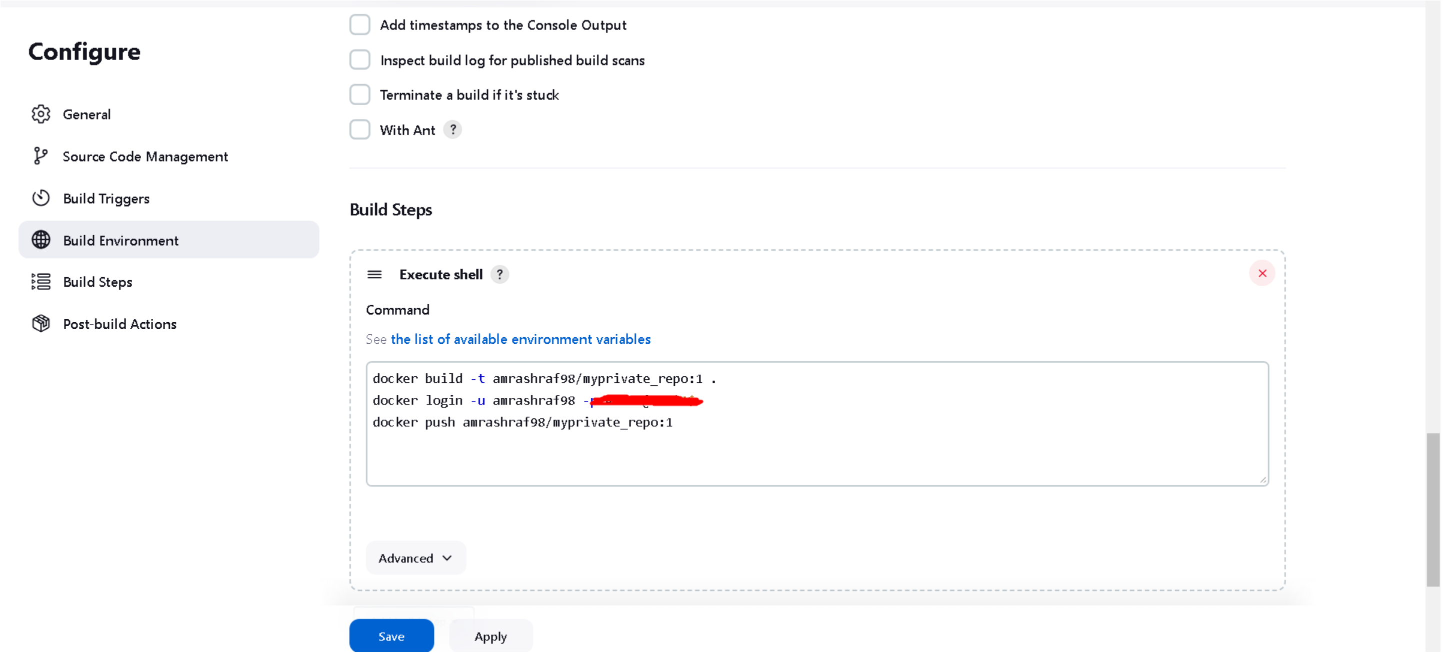
Repository URL

https://github.com/amrashraf-web/private_repo

Credentials

amrashraf-web/******

Branches to build



- □ Add timestamps to the Console Output
- □ Inspect build log for published build scans
- □ Terminate a build if it's stuck
- □ With Ant ?

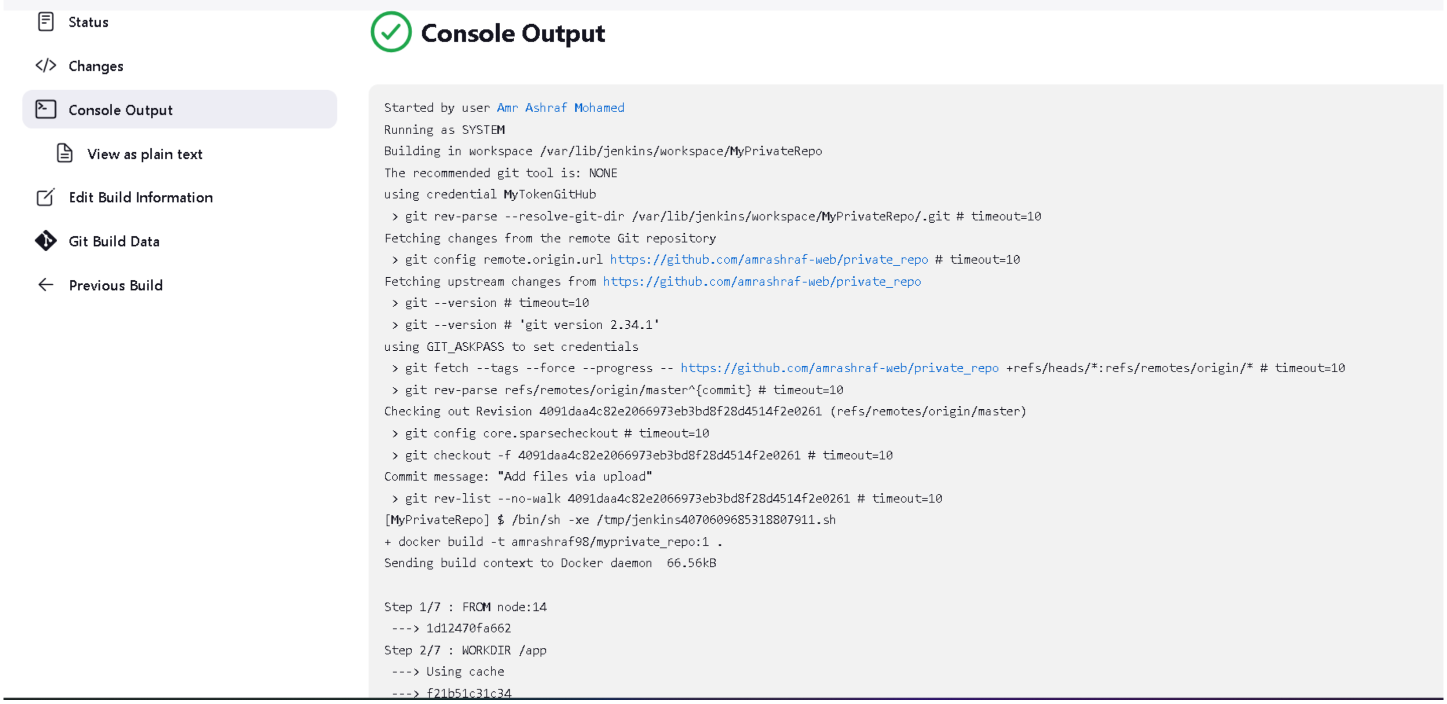
Build Steps

Execute shell ?

Command

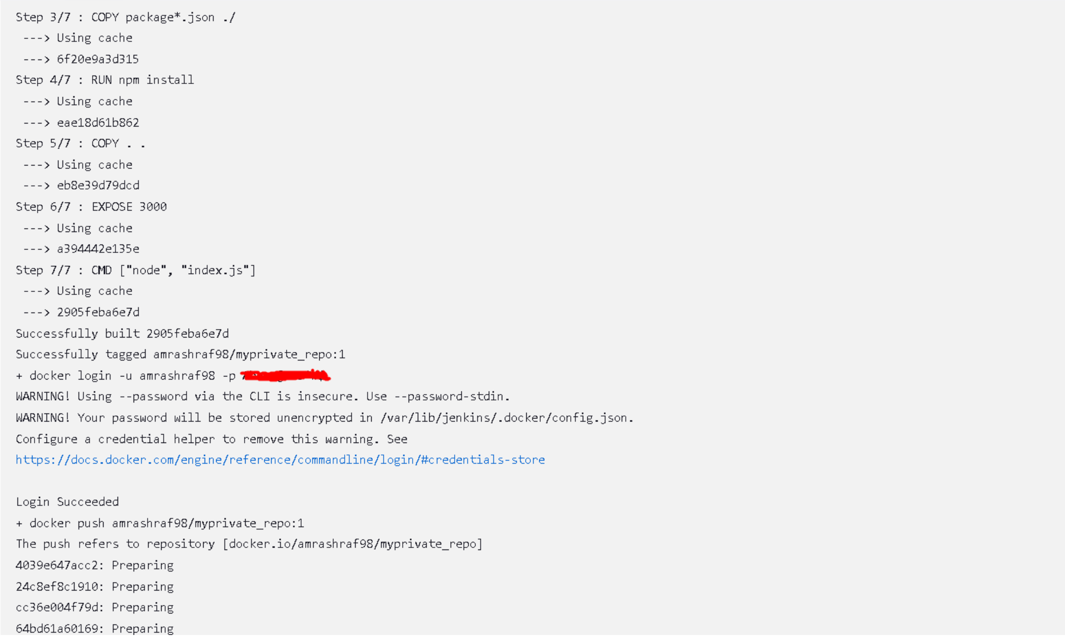
See the list of available environment variables

```
docker build -t amrashraf98/myprivate_repo:1 .
docker login -u amrashraf98 -[redacted]
docker push amrashraf98/myprivate_repo:1
```

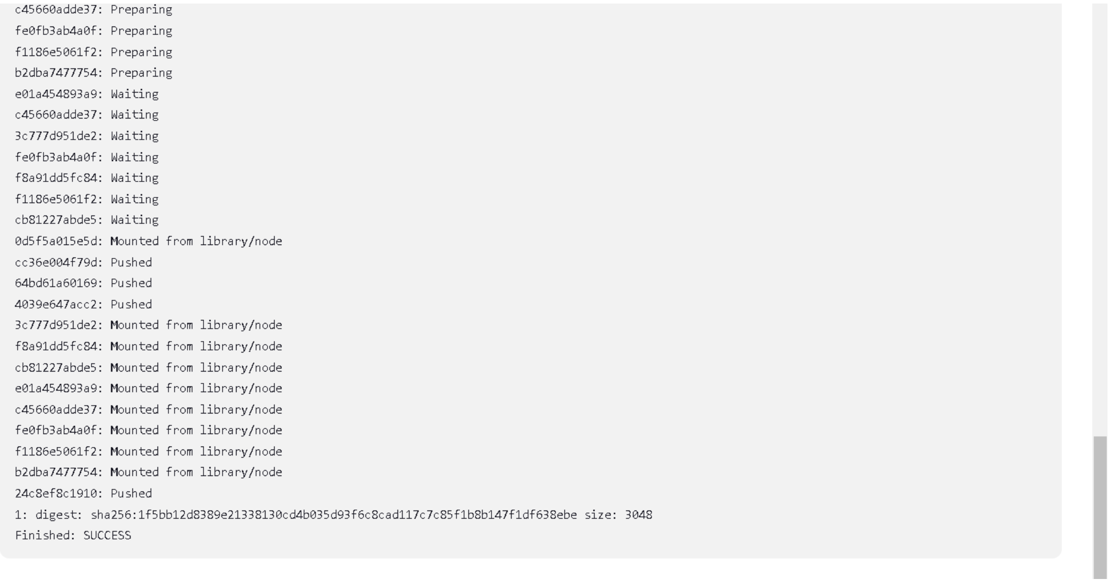


Console Output

```
Started by user Amr Ashraf Mohamed
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/MyPrivateRepo
The recommended git tool is: NONE
using credential MyTokenGitHub
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/MyPrivateRepo/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/amrashraf-web/private_repo # timeout=10
Fetching upstream changes from https://github.com/amrashraf-web/private_repo
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress -- https://github.com/amrashraf-web/private_repo +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 4091daa4c82e2066973eb3bd8f28d4514f2e0261 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 4091daa4c82e2066973eb3bd8f28d4514f2e0261 # timeout=10
Commit message: "Add files via upload"
 > git rev-list --no-walk 4091daa4c82e2066973eb3bd8f28d4514f2e0261 # timeout=10
[MyPrivateRepo] $ /bin/sh -xe /tmp/jenkins4070609685318807911.sh
+ docker build -t amrashraf98/myprivate_repo:1 .
Sending build context to Docker daemon  66.56kB

Step 1/7 : FROM node:14
 ---> 1d12470fa662
Step 2/7 : WORKDIR /app
 ---> Using cache
 ---> f21b51c31c34
```

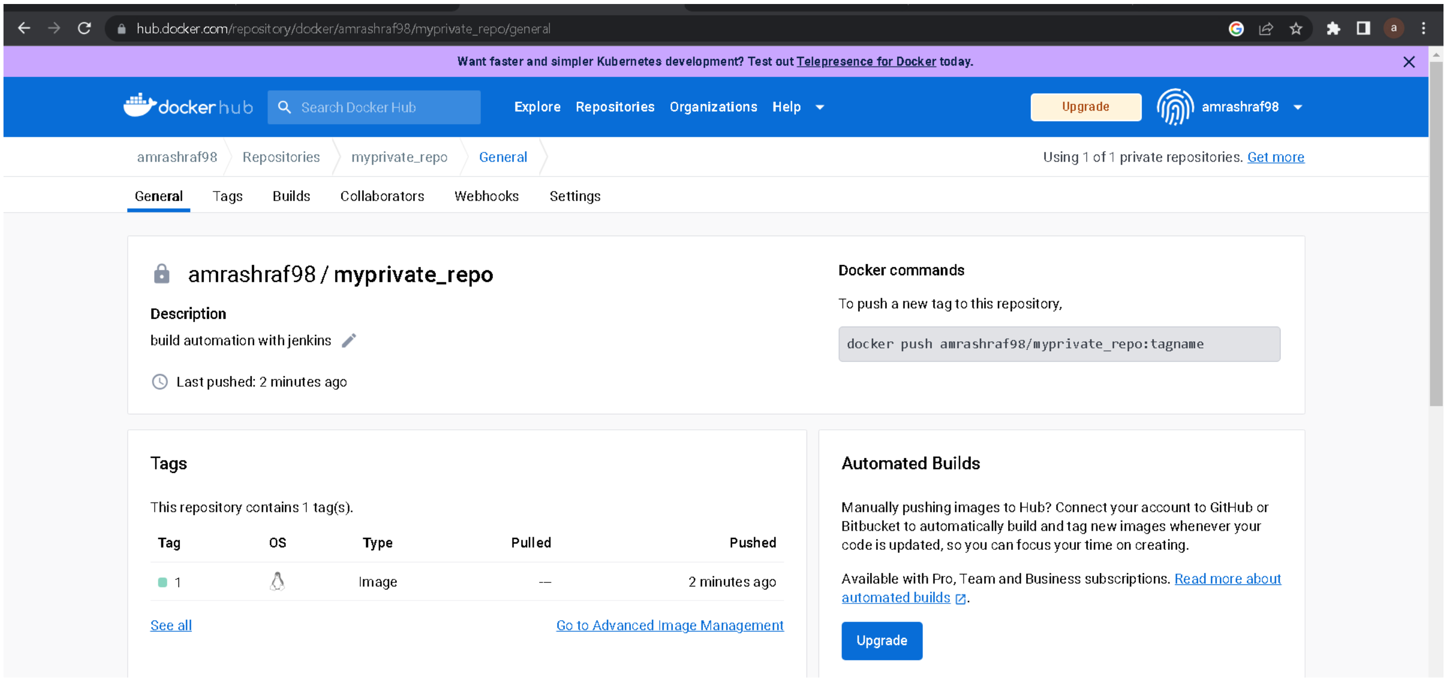


```
Step 3/7 : COPY package*.json ./
 ---> Using cache
 ---> 6f20e9a3d315
Step 4/7 : RUN npm install
 ---> Using cache
 ---> eae18d61b862
Step 5/7 : COPY . .
 ---> Using cache
 ---> eb8e39d79dcd
Step 6/7 : EXPOSE 3000
 ---> Using cache
 ---> a394442e135e
Step 7/7 : CMD ["node", "index.js"]
 ---> Using cache
 ---> 2905feba6e7d
Successfully built 2905feba6e7d
Successfully tagged amrashraf98/myprivate_repo:1
+ docker login -u amrashraf98 -p [redacted]
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
+ docker push amrashraf98/myprivate_repo:1
The push refers to repository [docker.io/amrashraf98/myprivate_repo]
4039e647acc2: Preparing
24c8ef8c1910: Preparing
cc36e004f79d: Preparing
64bd61a60169: Preparing
```



```
c45660adde37: Preparing
fe0fb3ab4a0f: Preparing
f1186e5061f2: Preparing
b2dba7477754: Preparing
e01a454893a9: Waiting
c45660adde37: Waiting
3c777d951de2: Waiting
fe0fb3ab4a0f: Waiting
f8a91dd5fc84: Waiting
f1186e5061f2: Waiting
cb81227abde5: Waiting
0d5f5a015e5d: Mounted from library/node
cc36e004f79d: Pushed
64bd61a60169: Pushed
4039e647acc2: Pushed
3c777d951de2: Mounted from library/node
f8a91dd5fc84: Mounted from library/node
cb81227abde5: Mounted from library/node
e01a454893a9: Mounted from library/node
c45660adde37: Mounted from library/node
fe0fb3ab4a0f: Mounted from library/node
f1186e5061f2: Mounted from library/node
b2dba7477754: Mounted from library/node
24c8ef8c1910: Pushed
1: digest: sha256:1f5bb12d8389e21338130cd4b035d93f6c8cad117c7c85f1b8b147f1df638ebe size: 3048
Finished: SUCCESS
```

And for docker



Want faster and simpler Kubernetes development? Test out Telepresence for Docker today.

Using 1 of 1 private repositories. Get more

amrashraf98 / **myprivate_repo**

Description

build automation with jenkins

Last pushed: 2 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push amrashraf98/myprivate_repo:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
🟩 1		Image	---	2 minutes ago

See all   Go to Advanced Image Management

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 with Pro, Team and Business subscriptions. Read more about automated builds.