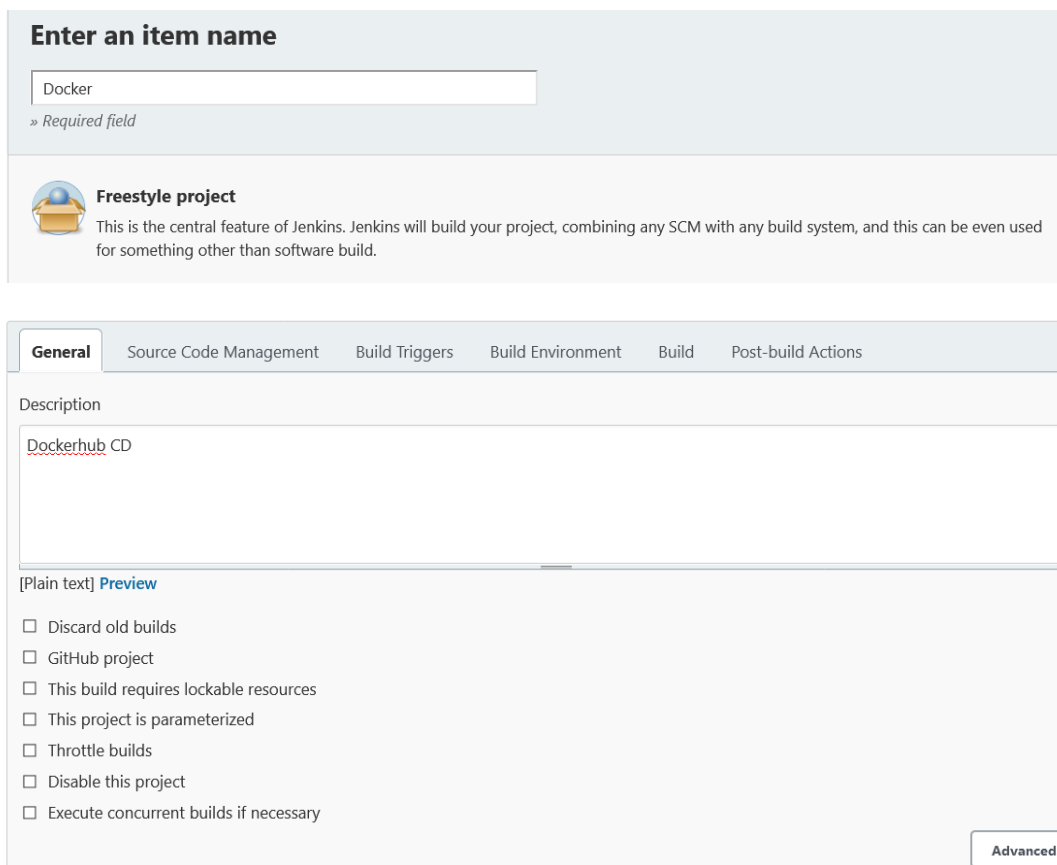


Group B
ENSE 375 - Project Step 4
April 5, 2021

Jenkins & Docker CD Process

Starting on Jenkins

First we create a free-style project again.




The image shows the Jenkins 'New Item' form. At the top, there's a section 'Enter an item name' with a text input field containing 'Docker' and a note '» Required field'. Below this is a 'Freestyle project' icon and description. The main form has tabs: 'General' (selected), 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. The 'General' tab shows a 'Description' field with 'Dockerhub CD'. Below the description is a '[Plain text] Preview' section with a list of checkboxes: 'Discard old builds', 'GitHub project', 'This build requires lockable resources', 'This project is parameterized', 'Throttle builds', 'Disable this project', and 'Execute concurrent builds if necessary'. Each checkbox has a corresponding help icon (a blue circle with a question mark). At the bottom right of the 'General' tab is an 'Advanced...' button.

Enter an item name

Docker

» 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.

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Description

Dockerhub CD

[Plain text] [Preview](#)

- ☐ Discard old builds ?
- ☐ GitHub project ?
- ☐ This build requires lockable resources ?
- ☐ This project is parameterized ?
- ☐ Throttle builds ?
- ☐ Disable this project ?
- ☐ Execute concurrent builds if necessary ?

Advanced...

Next we do similar steps to our CI process.
We connect our Jenkins project to a GitHub repo

☐ None
☒ Git

Repositories

Repository URL

https://github.com/JonBarVargas/ENSE375-groupB

Credentials

tristanhannibal@gmail.com/***** Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/main

Add Branch

We only care about the main, as this is the branch that passes our merge test of the previous process of CI.

To check if there are any updates daily, and build the project if there is, we must select the Poll SCM option. The @daily is a shortcut in Jenkins to specify we want to build daily.

☒ Poll SCM

Configure Jenkins to poll changes in SCM.

Note that this is going to be an expensive operation for CVS, as every polling requires Jenkins to scan the entire workspace and verify it with the server. Consider setting up a "push" trigger to avoid this overhead, as described in [this document](#)

Schedule

@daily

Would last have run at Monday, April 5, 2021 at 2:04:30 PM Central Daylight Time; would next run at Tuesday, April 6, 2021 at 2:04:30 PM Central Daylight Time.

☒ Ignore post-commit hooks

Next we package our project with Maven. This will run the tests, and then it will allow us to create a Docker image from what is packaged, if the tests pass.

Build

Invoke top-level Maven targets

X ?

Maven Version

mvn

Goals

package

Advanced...

After packaging, we installed the plugin:

[CloudBees Docker Build and Publish plugin](#)

<input checked="" type="checkbox"/>	<p>This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.</p>	1.3.3	Uninstall
-------------------------------------	--	-------	-----------

This allowed us to have a build step

Docker Build and Publish

X ?

Repository Name

ense375groupb/groupb-activity4

Tag

Docker Host URI

Server credentials

- none -

Add

Docker registry URL

Registry credentials






ense375groupb/*****


Add

Advanced...


This plugin defaults to Dockerhub, so all we need to do is point it to our Repo (ense375groupb/groupb-activity4) and give it our Dockerhub credentials.


This step will look for the Docker file in the workspace, and then build the Docker image. The image will be create and then pushed to Dockerhub


-  Build Now
-  Configure
-  Delete Project
-  Git Polling Log
-  Rename



Build History
trend ^

X


[#32 ense375groupb/groupb-activity4](#)
[Apr 5, 2021, 3:03 PM](#)


[#31 ense375groupb/groupb-activity4](#)
[Apr 5, 2021, 3:01 PM](#)


[#30 ense375groupb/groupb-activity4](#)
[Apr 5, 2021, 2:51 PM](#)


[#29 ense375groupb/groupb-](#)
[.. .. .](#)

```

5f70bf18a086: Layer already exists
38039e5ec148: Layer already exists
14ba0e532b9e: Layer already exists
c37631ce9afc: Layer already exists
d459b0fecb7b: Layer already exists
106aefa920ed: Layer already exists
2a2d81e590da: Layer already exists
1059b684144a: Layer already exists
b3577d595e75: Layer already exists
b1169e57b139: Layer already exists
3ee270f20d54: Layer already exists
4ef4adca5c3b: Layer already exists
latest: digest: sha256:79e94e01cd422d575e254ff6233cf08da8589112bc930e20125eb21b35471641 size: 3252
Finished: SUCCESS

```



Permalinks

- [Last build \(#32 ense375groupb/groupb-activity4\), 15 min ago](#)
- [Last stable build \(#32 ense375groupb/groupb-activity4\), 15 min ago](#)
- [Last successful build \(#32 ense375groupb/groupb-activity4\), 15 min ago](#)
- [Last failed build \(#28\), 1 hr 27 min ago](#)
- [Last unsuccessful build \(#28\), 1 hr 27 min ago](#)
- [Last completed build \(#32 ense375groupb/groupb-activity4\), 15 min ago](#)

Thus the deployment is complete.