Lab 5: Using Docker Agent with Jenkins

Author: Gourav Shah

Publisher: School of Devops

Version : v2024.05.03.01

Objectives:

- You will learn how to prepare Jenkins environment to build with a docker agent
- Refactor Jenkinsfile with docker based agents

Steps:

- Read : Using Docker with Pipeline
- Refactor the Jenkinsfile for users app using the following reference

Smoke Testing Jenkins Integration with Docker

Create a test pipeline to check whether docker is been integrated with jenkins and to find out if you could use docker agent to run a pipeline job.

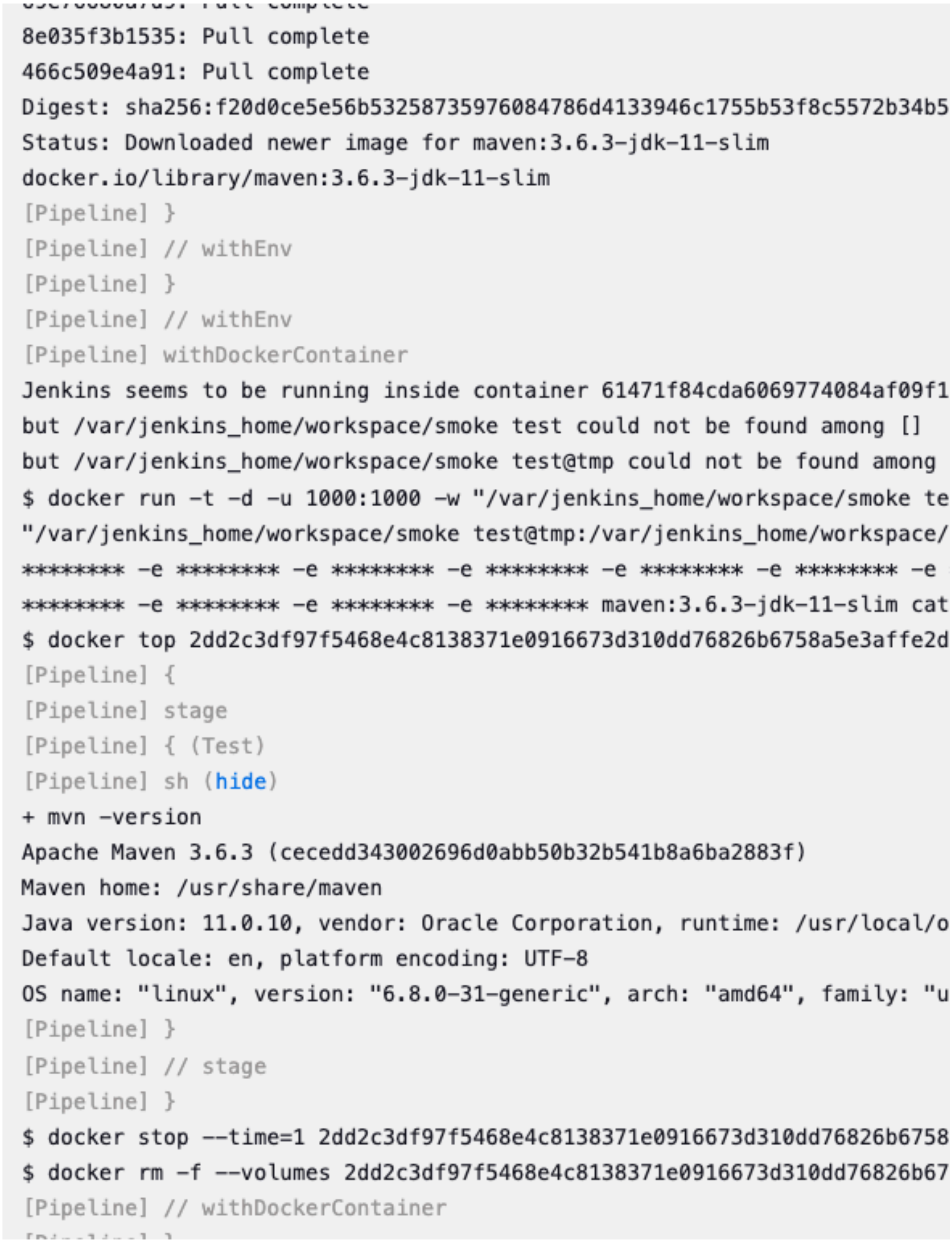
Using classic Jenkins UI, from the top jenkins page, create `docker-smoke-test` pipeline job. On the configuration page, add the following test code in jenkins pipeline script, save the configuration and build it.

```
pipeline {
    agent {
        docker { image 'maven:3.9.6-eclipse-temurin-17-alpine' }
    }
    stages {
        stage('Test') {
            steps {
                sh 'mvn -version'
```

```
                    }
                }
            }
        }
```

Run the job, and check if that worked? If not, you would have to install the following docker plugins (install without restart). If it worked, proceed to configure the agent as below.

Following is a console log from a successful run where you see jenkins pulling an image, launching a container to run `mvn version`

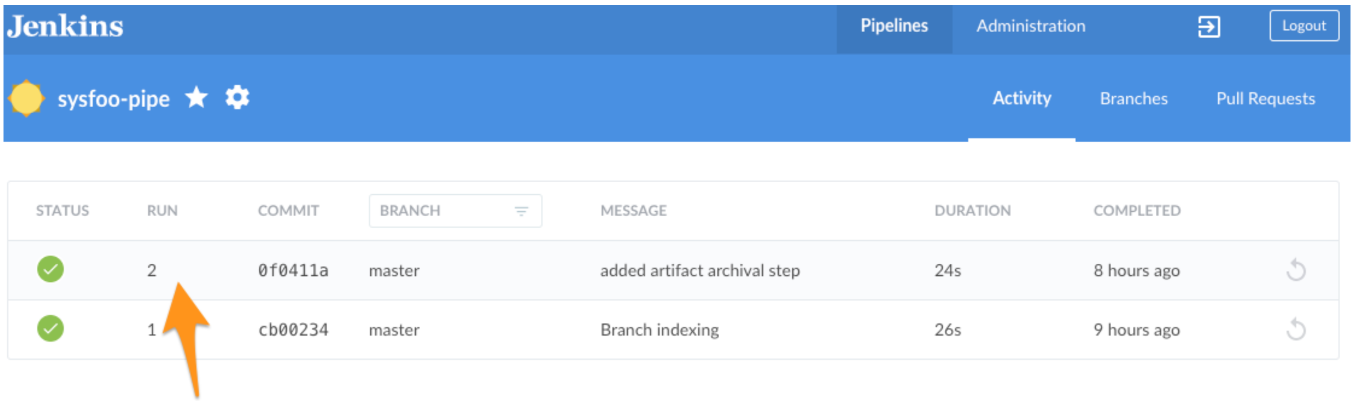


```
8e035f3b1535: Pull complete
466c509e4a91: Pull complete
Digest: sha256:f20d0ce5e56b532587359760847 86d4133946c1755b53f8c5572b34b5
Status: Downloaded newer image for maven:3.6.3-jdk-11-slim
docker.io/library/maven:3.6.3-jdk-11-slim
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] withDockerContainer
Jenkins seems to be running inside container 61471f84cda6069774084af09f1
but /var/jenkins_home/workspace/smoke test could not be found among []
but /var/jenkins_home/workspace/smoke test@tmp could not be found among
$ docker run -t -d -u 1000:1000 -w "/var/jenkins_home/workspace/smoke te
"/var/jenkins_home/workspace/smoke test@tmp:/var/jenkins_home/workspace/
******** -e ******** -e ******** -e ******** -e ******** -e ******** -e
******** -e ******** -e ******** -e ******** maven:3.6.3-jdk-11-slim cat
$ docker top 2dd2c3df97f5468e4c8138371e0916673d310dd76826b6758a5e3affe2d
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] sh (hide)
+ mvn -version
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /usr/share/maven
Java version: 11.0.10, vendor: Oracle Corporation, runtime: /usr/local/o
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-31-generic", arch: "amd64", family: "u
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
$ docker stop --time=1 2dd2c3df97f5468e4c8138371e0916673d310dd76826b6758
$ docker rm -f --volumes 2dd2c3df97f5468e4c8138371e0916673d310dd76826b67
[Pipeline] // withDockerContainer
```

Configuring Docker Agent

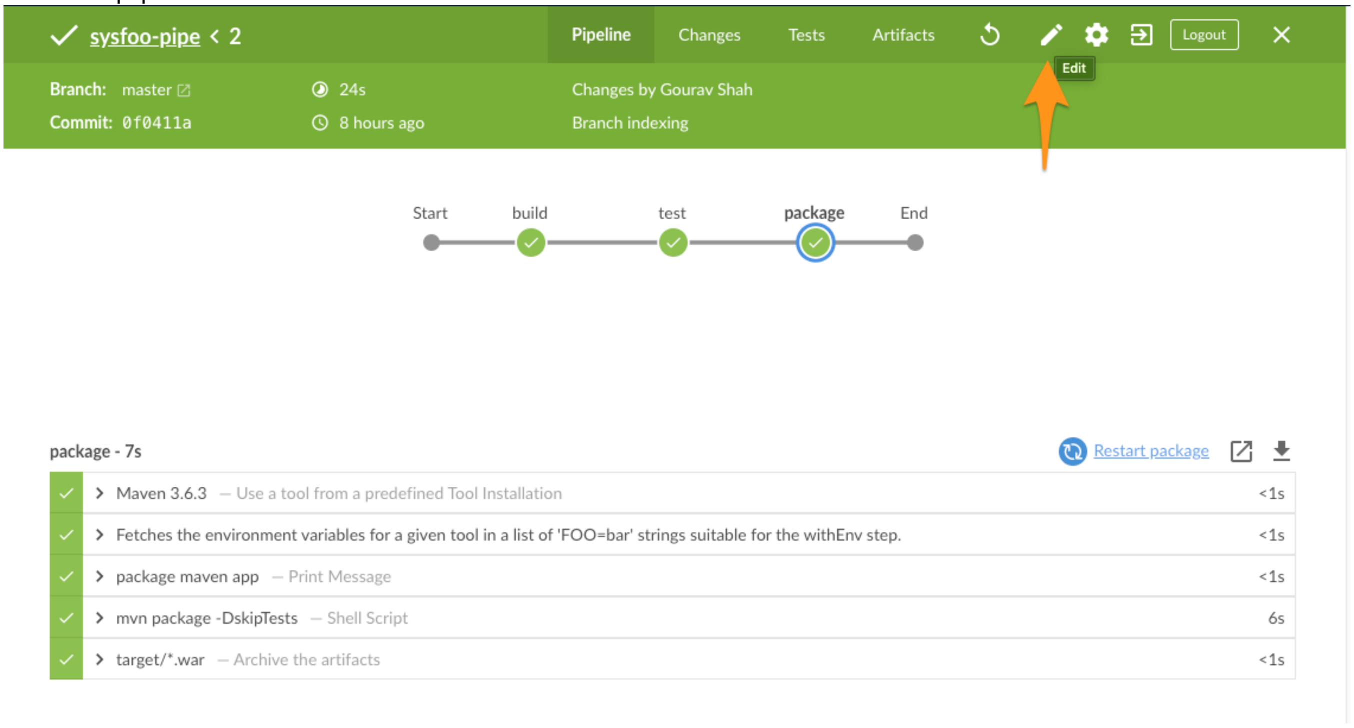
To update the pipeline with docker agent is very simple. Blue Ocean just makes it a breeze. To use docker agent,

From Blue Ocean UI, head to one of the pipeline runs for **sysfoo-pipe**,



STATUS	RUN	COMMIT	BRANCH	MESSAGE	DURATION	COMPLETED
✓	2	0f0411a	master	added artifact archival step	24s	8 hours ago
✓	1	cb00234	master	Branch indexing	26s	9 hours ago

Edit the pipeline with Blue Ocean

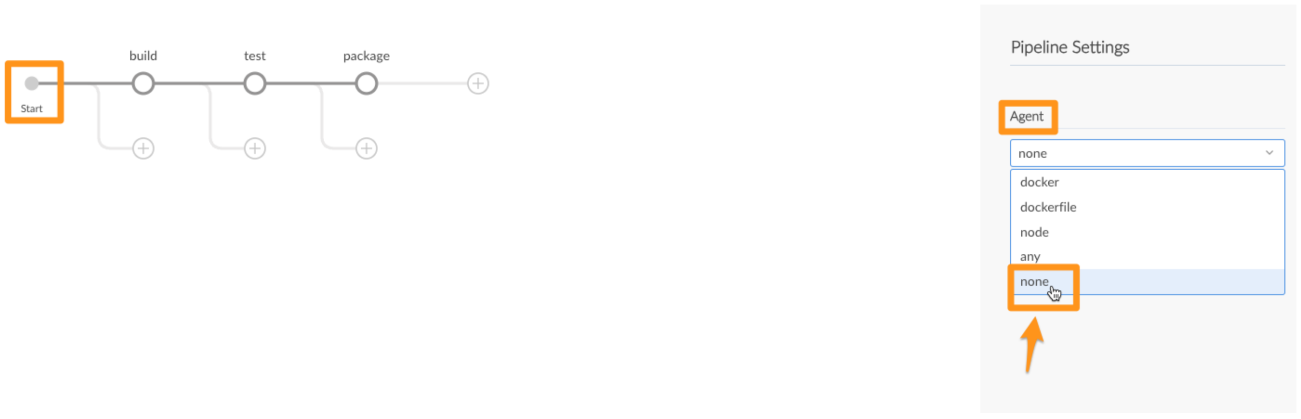


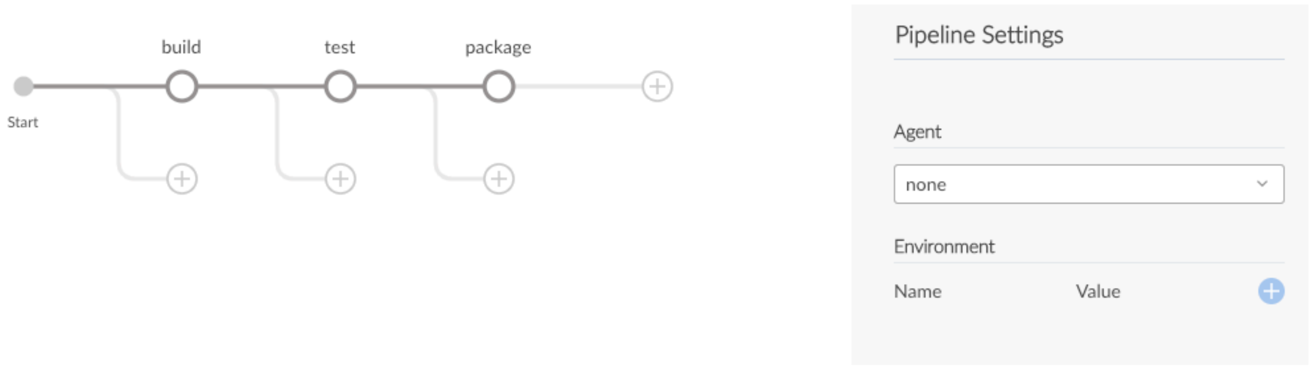
Branch: master
Commit: 0f0411a
24s
8 hours ago
Changes by Gourav Shah
Branch indexing

package - 7s

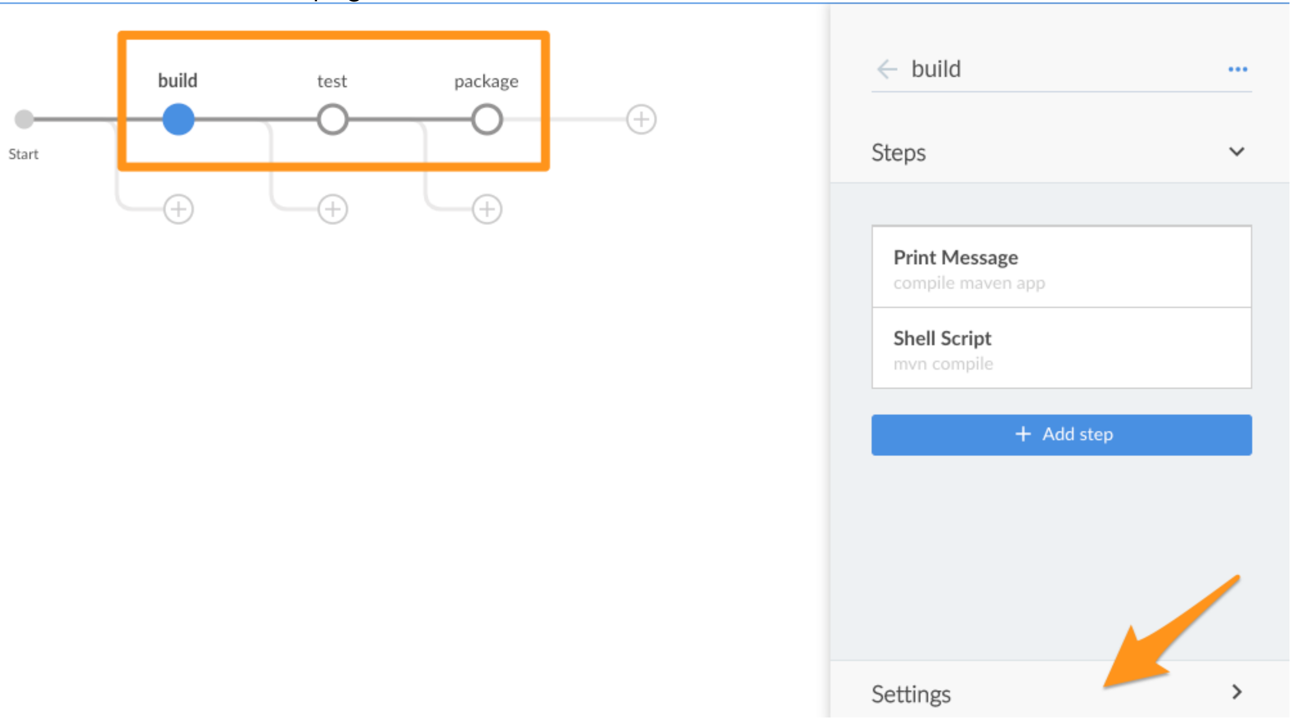
✓	Maven 3.6.3 — Use a tool from a predefined Tool Installation	<1s
✓	Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.	<1s
✓	package maven app — Print Message	<1s
✓	mvn package -DskipTests — Shell Script	6s
✓	target/*.war — Archive the artifacts	<1s

From the pipeline settings (Global Settings) dropdown, set the agent to **none**. This will ensure the agent configurations will be read from every stage.

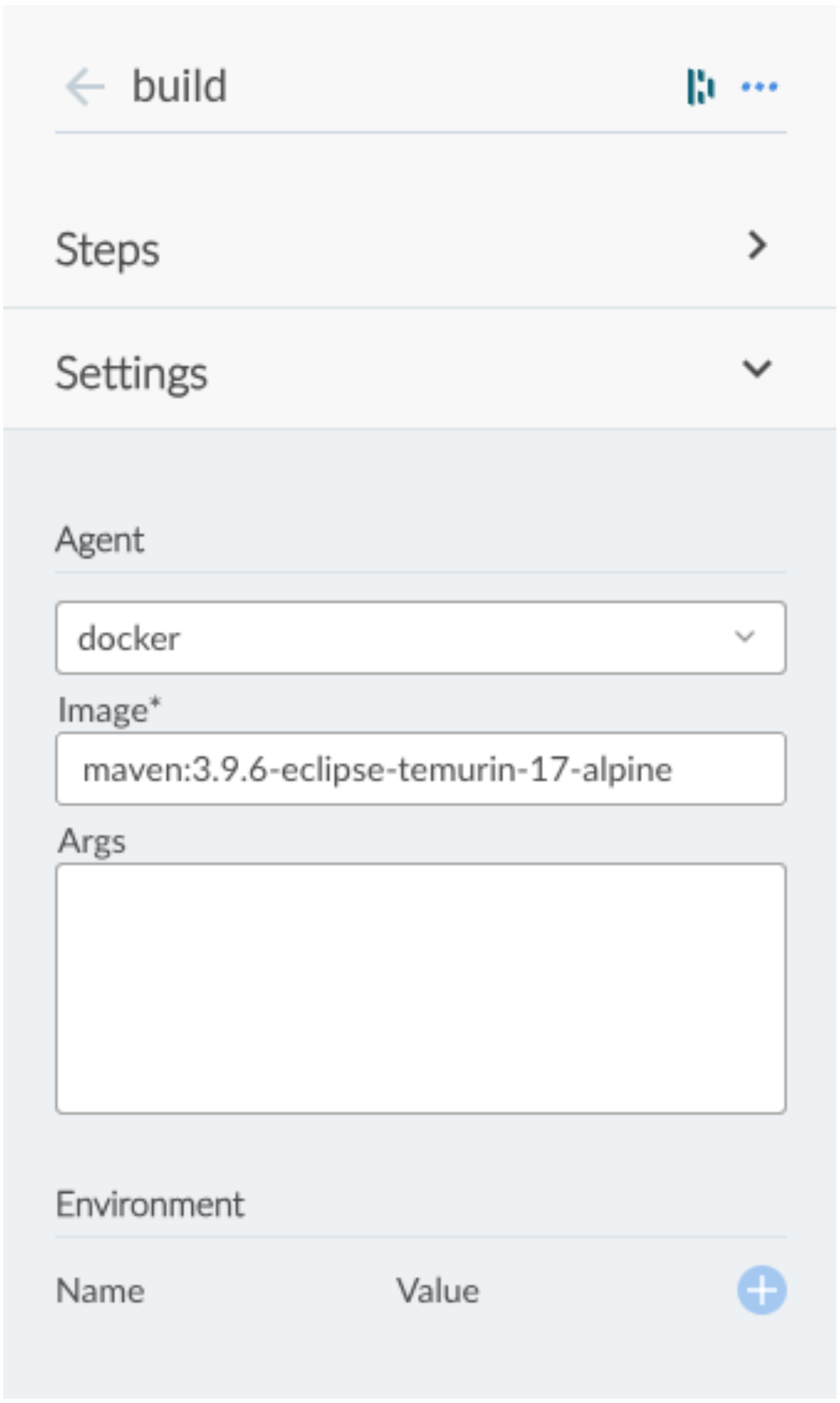




Now, for each of the three stages i.e. build, test and package, go the **settings** option at the bottom of the page.



Change agent from **None** to **Docker** and provide the same image that you are going to use to build these stages with e.g. `maven:3.9.6-eclipse-temurin-17-alpine`



build

Steps

Settings

Agent

docker

Image*

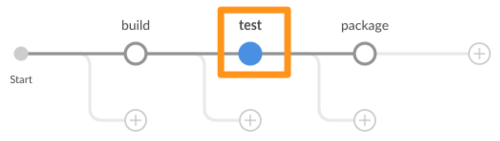
maven:3.9.6-eclipse-temurin-17-alpine

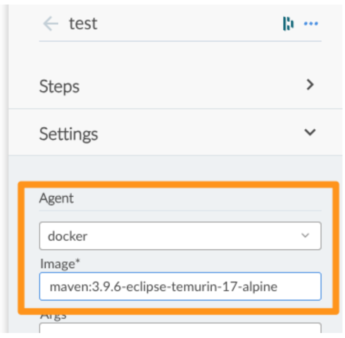
Args

Environment

Name Value

```
STOP:  make sure you have updated the configuration for all three stages before
proceeding. Else the pipeline would fail to run.
```



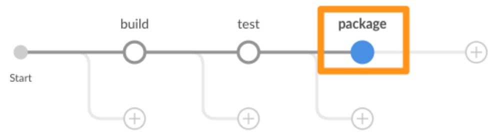


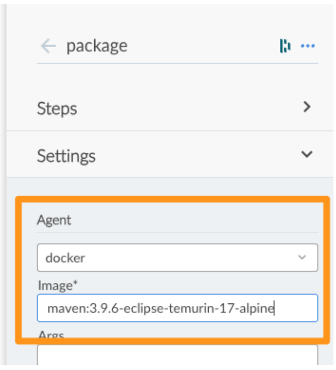
Agent

docker

Image*

maven:3.9.6-eclipse-temurin-17-alpine





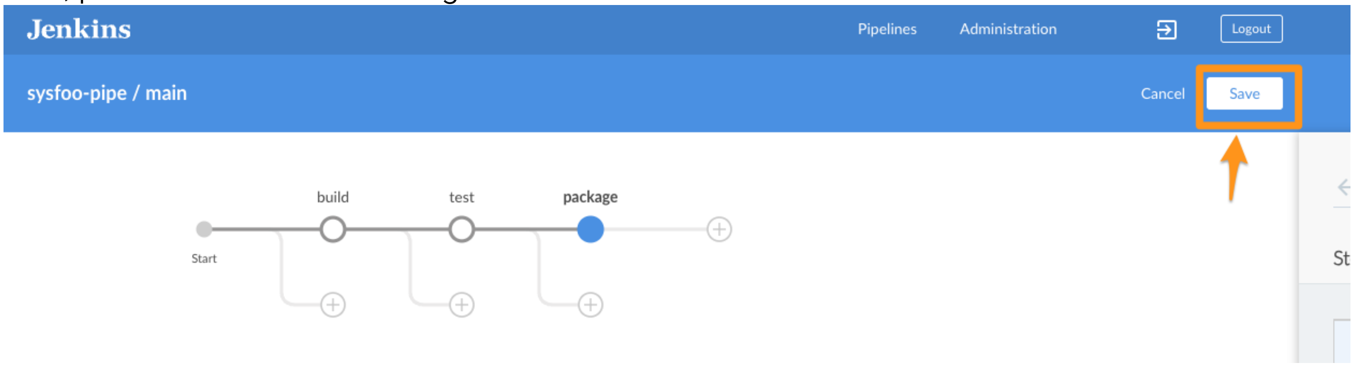
Agent

docker

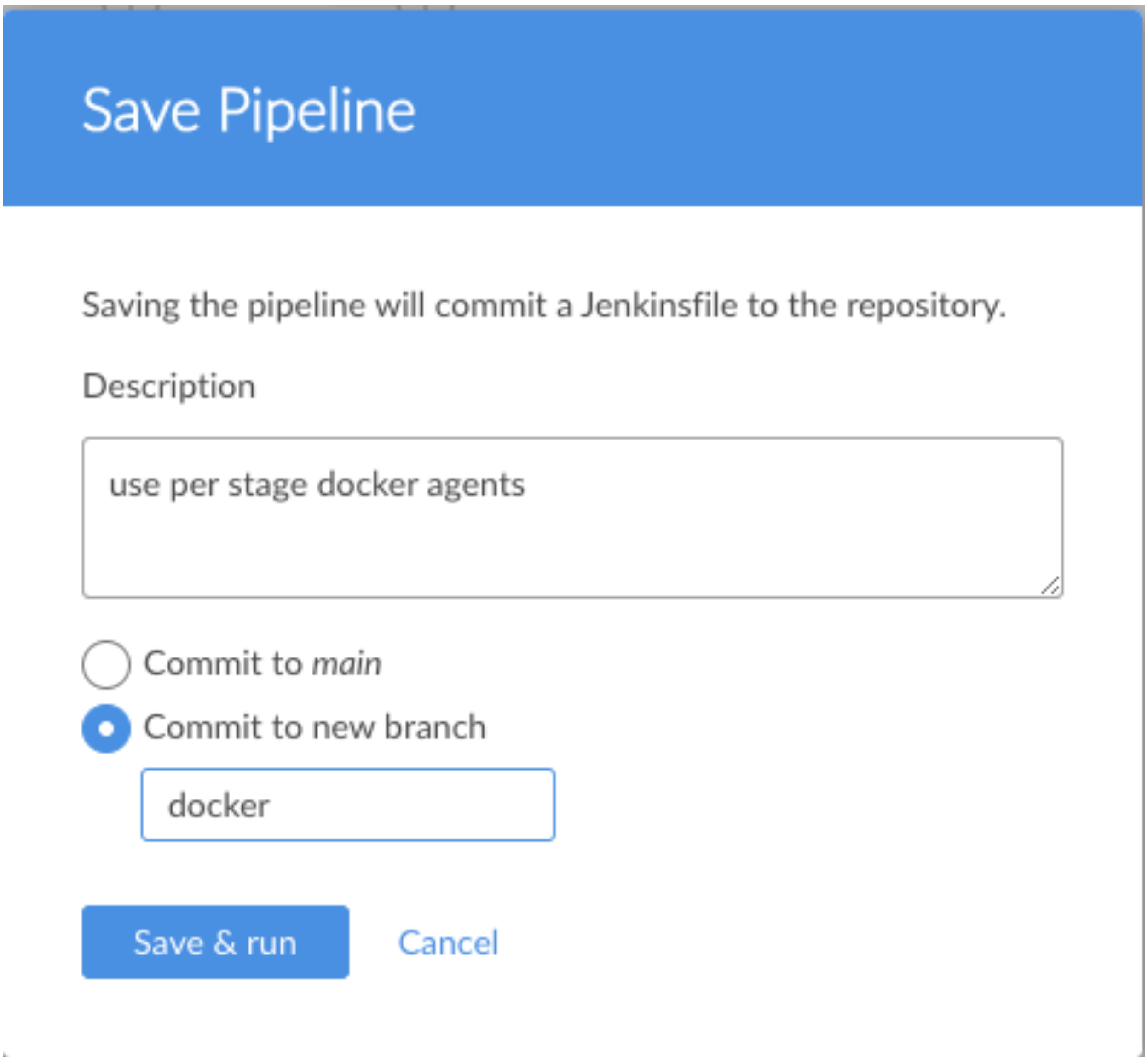
Image*

maven:3.9.6-eclipse-temurin-17-alpine

Now, proceed to save these changes.



Add a description and commit with a new branch e.g. `docker` as **master** branch is locked (protected) and does not allow any direct check ins.


Save Pipeline

Saving the pipeline will commit a Jenkinsfile to the repository.

Description: use per stage docker agents

Commit to *main* [✓ Commit to new branch]: docker

To validate that docker agent is being used, get inside the DIND continuer and watch

```
cd bootcamp/jenkins

docker-compose exec  docker sh

watch docker ps
```

You would notice that,

- While the pipeline runs, you see a container created for every stage with the image defined as part of the pipeline configurations.

```
Every 2.0s: docker ps
2020-10-20 16:38:47

CONTAINER ID        IMAGE                          COMMAND                  CREATED
STATUS              PORTS               NAMES
10e236f59d7c        maven:3.9.6-eclipse-temurin-17-alpine   "/usr/local/bin/
mvn-…"   9 seconds ago       Up 8 seconds
relaxed_robinson
```

- After each pipeline stage is finished, you would also notice that this container is automatically deleted.

This validates the point that with docker agent, a new, clean, disposable environment can be created for every single instance of the pipeline run.

You could exit the watch command with `ctrl + c` followed by `exit` command to come out of the DIND environment.

#cicd/labsv3