# Jenkins Cluster in Docker Swarm

## Introduction

The goal is to setup a Jenkins master and several slaves running inside Docker swarm. The slaves should automatically find and attach themselves to the Jenkins master. The Jenkins master should be preconfigured with a series of plugins for all the basic stuff, such as pipeline, github and more. In this way there should be no configuring the master every time it starts.

## Compose file

Below is the resultant compose file to create the cluster from the built images. At the bottom of the file are several secrets that need to be created in the swarm so that the Jenkins master image will create the user and password inside. It also needs a secret for the Jenkins slaves to be able to connect to the master.

Another thing to note is that the Jenkins-slave service has the docker socket mapped as a volume. This enables the jenkins-slave service to issue docker commands and have them executed by the hosts docker daemon.

version: '3.4'

networks:

jenkins\_net:

secrets:

jenkinsUser:

external: true

jenkinsPassword:

external: true

jenkinsSwarm:

external: true

services:

jenkins-master:

image: joncatlin/jenkins-swarm-master

volumes:

- /mount/VM-File-Storage/jenkins-master:/var/jenkins\_home

ports:

- "8080:8080"

networks:

- jenkins\_net

secrets:

- source: jenkinsUser

target: jenkinsUser

uid: '1000'

mode: 0440

- source: jenkinsPassword

target: jenkinsPassword

uid: '1000'

mode: 0440

jenkins-slave:

image: joncatlin/jenkins-swarm-slave

volumes:

- /var/run/docker.sock:/var/run/docker.sock

networks:

- jenkins\_net

secrets:

- source: jenkinsSwarm

target: jenkinsSwarm

mode: 0440

deploy:

replicas: 3

update\_config:

parallelism: 1

delay: 30s

restart\_policy:

condition: on-failure

delay: 5s

max\_attempts: 3

window: 20s

# The following lines are the ones used to set up the secrets for testing

# echo admin | docker secret create jenkinsUser -

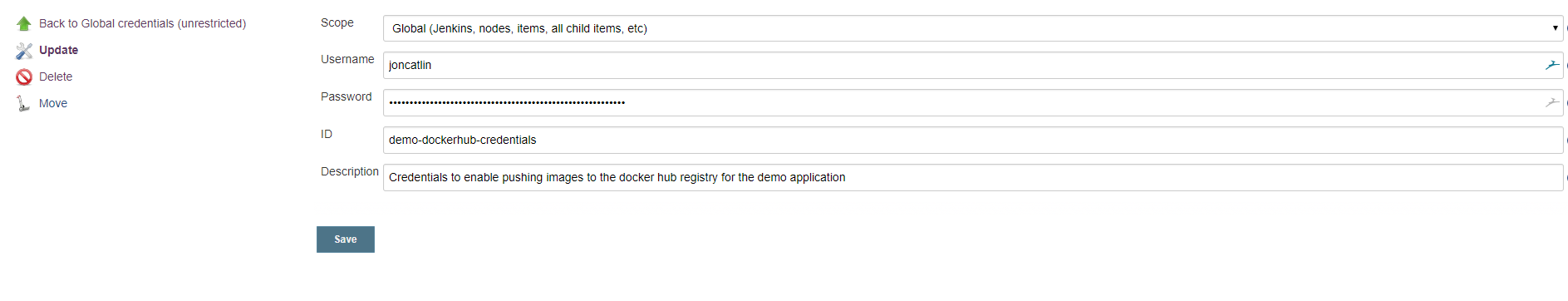
# echo admin | docker secret create jenkinsPassword -

# echo -master http://jenkins-master:8080 -password admin -username admin | docker secret create jenkinsSwarm -

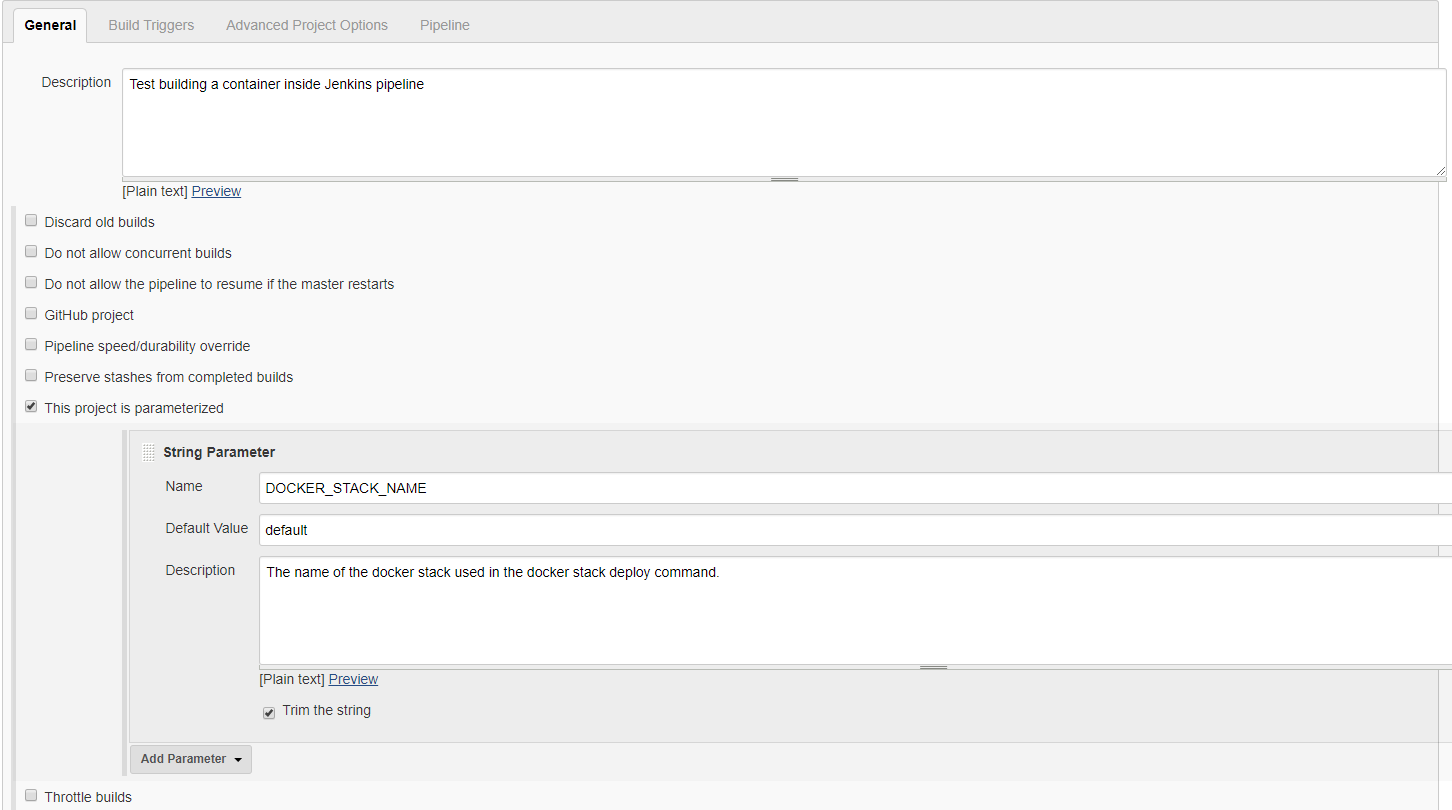
## Configuration

Once the Jenkins master container is running and the logs (docker service logs -f …) show that it is stable. Connect to the container using port 8080 on the swarm and add the following configuration.

The first configuration needed is to define a credential (username and password) that allows it to connect to the docker hub registry and be able to push a built image of the demo container application.



The next configuration task is to build a Jenkins pipeline job with the following parameters. The job has a single parameter that is the name of the docker stack to use when issuing the ‘docker stack deploy –composefile …’ command.



The second part of the job’s configuration should look like the image below. It shows that the pipeline should be run from the script that has been checked into the repo root and called Jenkinsfile.

