




CI/CD Pipeline with Docker

Project ID - 2, Team ID - 13



Problem Statement

- Current CI/CD solutions follow multi-tiered environments approach - development, test, staging and production.
- Each of these environments are managed independently of each other. Hence, each of these environments may have different configurations different library versions or even different Operating Systems.
- This leads to the popular problem known as *“it works on my machine”* syndrome where an application that works on one environment stops working on some other due to the above mentioned problem

Problems with the current CI/CD systems

- Difficulty supporting diverse language stacks and tooling
- Slow provisioning and setup of build and test environments
- Low throughput of jobs and software shipped to stage or production
- Inconsistencies between environments

Solution using docker

A docker based solution has the following advantages over the traditional ways:

- Eliminate system and language conflicts by isolating in containers
- Run more jobs faster
- Ship more software
- Standardized yet flexible environment

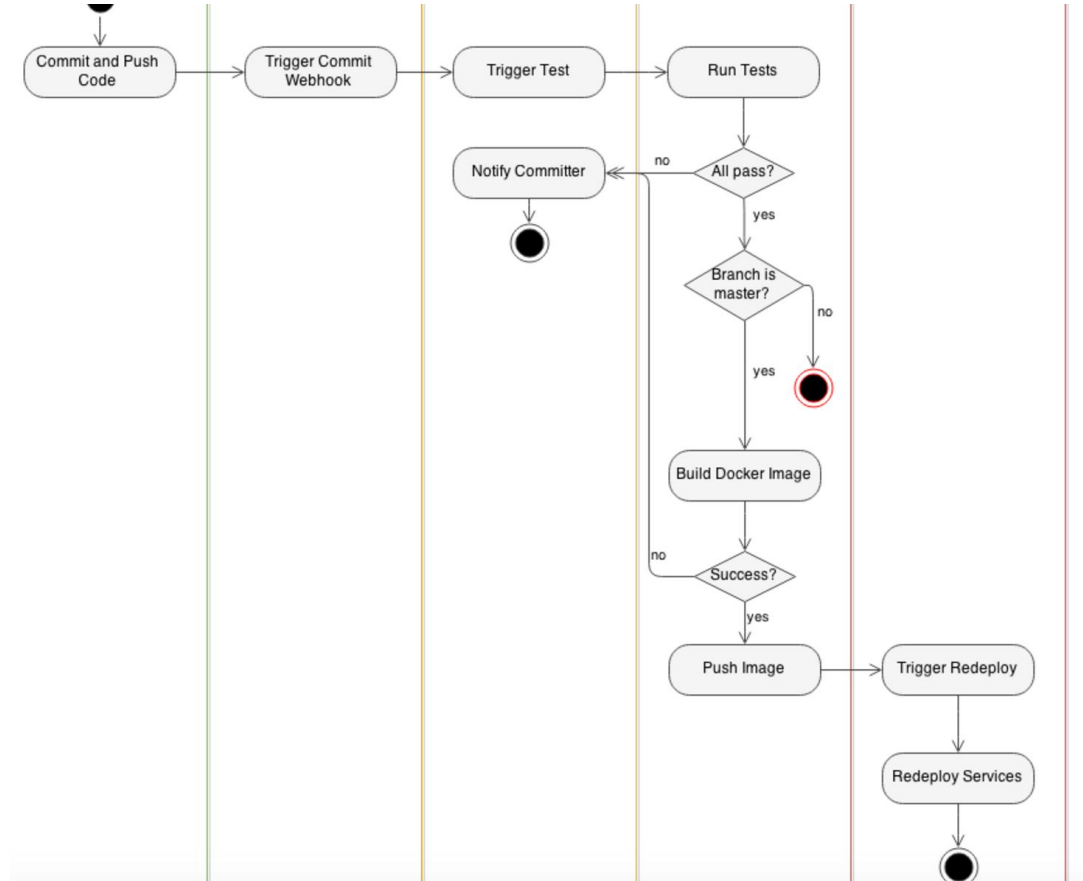
Overview

- The basic idea for the project is to try to create a build system which will use Docker for implementing continuous integration (CI) / Continuous Deployment Pipeline.
- A git based commit should be used for starting of a build for a docker image which would then be run and provisioned in a Virtual Machine.
- After every commit a series of test cases is then run on the code to ensure the correctness of the code.
- After all the test-cases pass, the image gets updated on docker-hub registry, and a VM gets provisioned which can then run the software directly (after pulling the image from the docker-hub).

Approach

- Use a docker based solution for this problem of ensuring fast continuous delivery and deployment.
- The use of docker based approach eliminates the system and language conflicts by isolating the things in a containers.
- The use of container will ensure a smooth integrated environment throughout. It will also make the process more streamlined and faster.
- For every push a webhook will notify of the changes in the code. For every such notification the DockerFile and the code will be pulled and a docker image will be build.
- Tests will be run on the slave nodes and if all the test-cases pass, then a VM will be provisioned for the same.

Workflow





Implementation



Commit Message

```
5 # Changes to be committed:  
6 #   new file:   test1  
7 #  
8 # Prepend commit with [NO CI] to prevent automated testing.
```

Post - Commit

```
Build job sent for commit 15a01c521140bf5a1dd3a2b56af71e7b2870c1d8  
  
[master 15a01c5] asdf  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 test1
```

Success Notification

Sun Nov 22, 11:09 PM

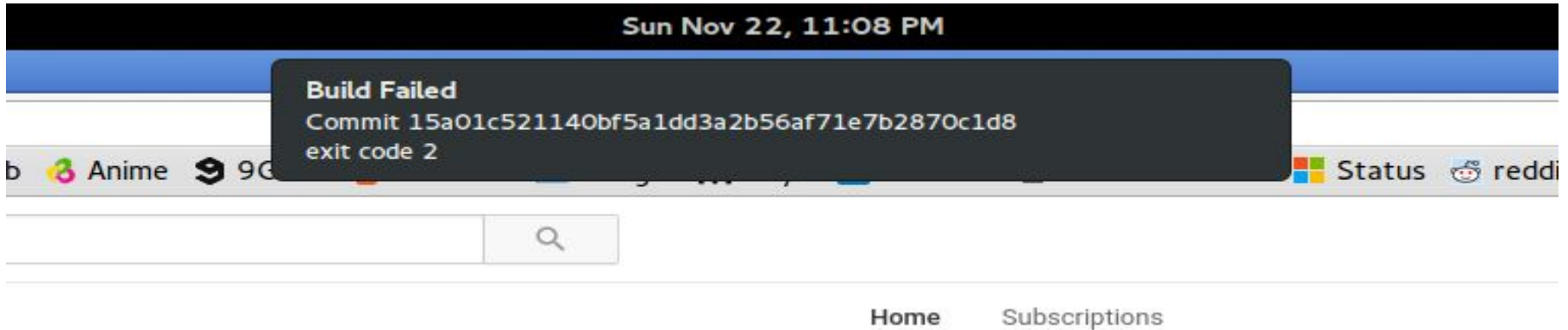
Build Successful

Commit Oe96e20e27aeee9989b79d85b012b4b842d61a85

Anime 9GAG Research Google M DayT LinkedIn GeeksforGeeks Status



Failure Notification



Timeline of the containers (and their status)

```
kushal@ubuntu:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2709432b3cbc	ubuntu	"/bin/bash"	14 seconds ago	Up 14 seconds		trusting_dijkstra

```
kushal@ubuntu:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2709432b3cbc	ubuntu	"/bin/bash"	4 minutes ago	Exited (1) 24 seconds ago		trusting_dijkstra

```
kushal@ubuntu:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2709432b3cbc	ubuntu	"/bin/bash"	5 minutes ago	Exited (1) About a minute ago		trusting_dijkstra

```
kushal@ubuntu:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0daaf54d1346	ubuntu	"/bin/bash"	2 minutes ago	Up 2 minutes		serene_fermi
2709432b3cbc	ubuntu	"/bin/bash"	8 minutes ago	Exited (1) 4 minutes ago		trusting_dijkstra

```
kushal@ubuntu:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0daaf54d1346	ubuntu	"/bin/bash"	3 minutes ago	Exited (0) 24 seconds ago		serene_fermi
2709432b3cbc	ubuntu	"/bin/bash"	9 minutes ago	Exited (1) 5 minutes ago		trusting_dijkstra

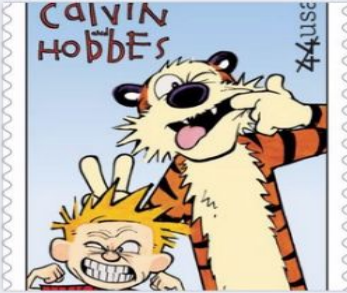
```
kushal@ubuntu:~$
```

Before pushing updated image

https://hub.docker.com/u/kushalsingh007/

Dashboard Explore Organizations Search Create kushalsingh007

Repos Stars



kushalsingh007
Kushal Singh

Joined May 2015





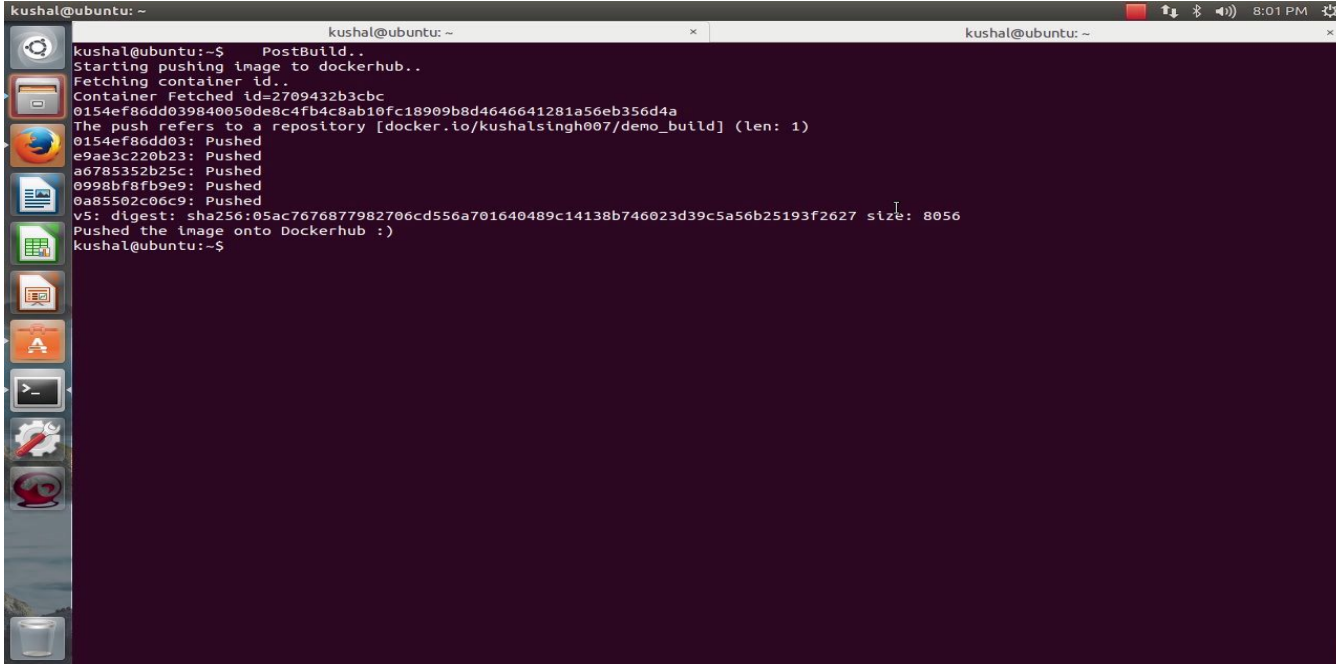
 kushalsingh007/dynamic_v1 public	0 STARS	9 PULLS	DETAILS
 kushalsingh007/proxy_ubuntu public	0 STARS	3 PULLS	DETAILS
 kushalsingh007/build public	0 STARS	0 PULLS	DETAILS
 kushalsingh007/go public	0 STARS	2 PULLS	DETAILS

Image getting pushed to Docker-hub successfully

A terminal window titled 'kushal@ubuntu: ~' with a dark purple background. The window shows the output of a Docker build and push command. The output includes the container ID, the repository name, and a list of layers being pushed. The final line shows the image is pushed onto Dockerhub.

```
kushal@ubuntu: ~  
kushal@ubuntu:~$ PostBuild..  
Starting pushing image to dockerhub..  
Fetching container id..  
Container Fetched id=2709432b3cbc  
0154ef86dd039840050de8c4fb4c8ab10fc18909b8d4646641281a56eb356d4a  
The push refers to a repository [docker.io/kushalsingh007/demo_build] (len: 1)  
0154ef86dd03: Pushed  
e9ae3c220b23: Pushed  
a6785352b25c: Pushed  
0998bf8fb9e9: Pushed  
0a85502c06c9: Pushed  
v5: digest: sha256:05ac7676877982706cd556a701640489c14138b746023d39c5a56b25193f2627 size: 8056  
Pushed the image onto Dockerhub :)  
kushal@ubuntu:~$
```

After Pushing the updated image (demo_build)

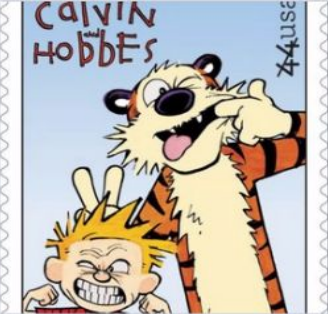
https://hub.docker.com/u/kushalsingh007/

Dashboard Explore Organizations

Search






Create kushalsingh007

Repos Stars

 44 users

kushalsingh007
Kushal Singh

Joined May 2015

 kushalsingh007/dynamic_v1 public	0 STARS	9 PULLS	> DETAILS
 kushalsingh007/proxy_ubuntu public	0 STARS	3 PULLS	> DETAILS
 kushalsingh007/build public	0 STARS	0 PULLS	> DETAILS
 kushalsingh007/go public	0 STARS	2 PULLS	> DETAILS
 kushalsingh007/demo_build public	0 STARS	2 PULLS	> DETAILS