Harshini Mitta

CSC 9010-033: Containers & Microservices

Summer 2017

Technical Specification :

- Java
  - Spring Boot
- Basic HTML
- Basic CSS
- IDE
  - Eclipse Neon
- GitHub
- Maven
- Docker

All about the project

- A simple web service
- A home page
- A simple click to details page
- A simple query parameter to greet
- Automated Deployment

Dockerfile

```
FROM openjdk:8
ADD target/docker-spring-boot.jar docker-spring-boot.jar
EXPOSE 9999/tcp
ENTRYPOINT ["java", "-jar", "docker-spring-boot.jar"]
```

Building and Pushing the Image

- Build the Maven project
- Build Container image
  - In Docker terminal switch to the project location
  - $ docker build –f Dockerfile –t spring-boot-service .
- To run docker image in a container
  - $ docker run –p <host port>:<container port> <image>
  - $ docker run –p 9999:9999 spring-boot-service

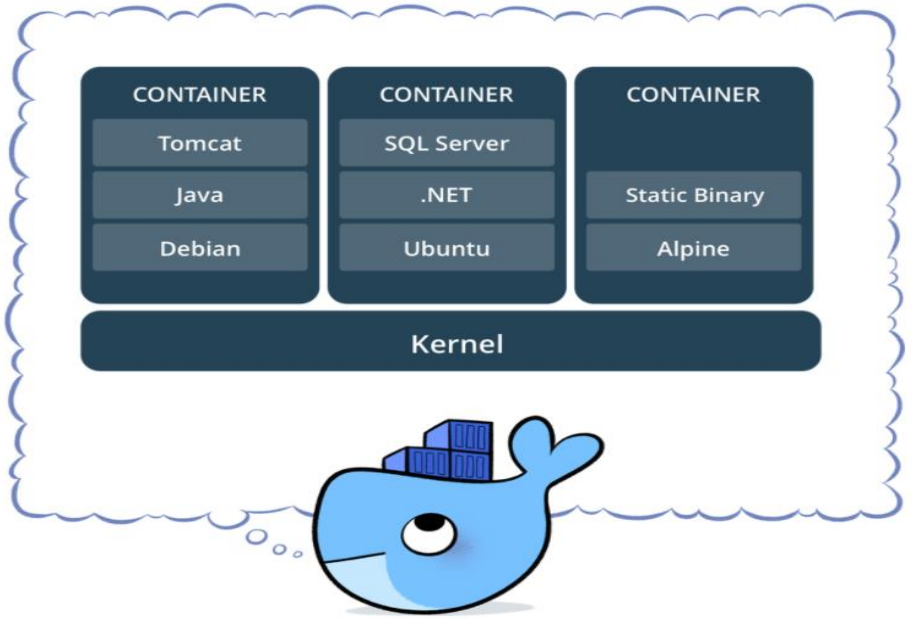
Home Page


192.168.99.101:9999
Welcome To my Containers & Microservices Project
docker
Build
Ship
Run
To know more click here

Details Page

192.168.99.101:9999/welcome

Hello, World!

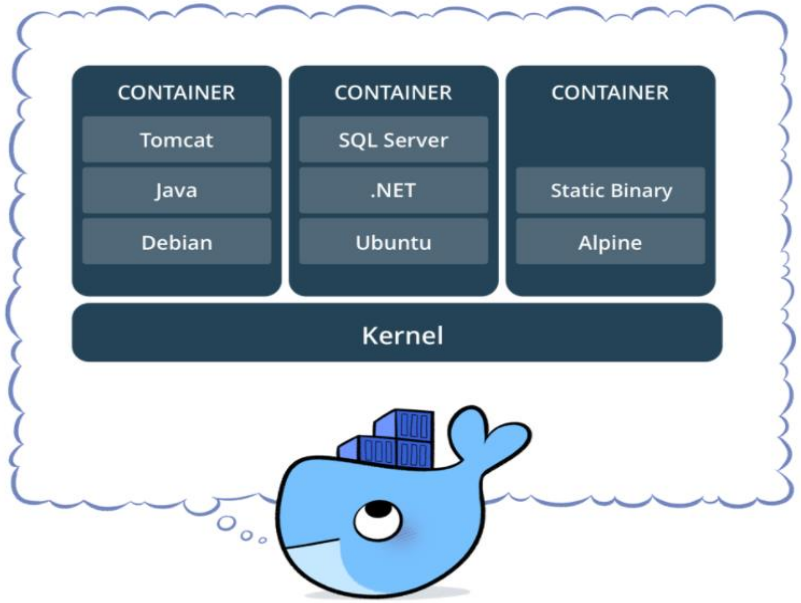


Container Image

Query Param

192.168.99.101:9999/welcome?name=Harshini

Hello, Harshini!

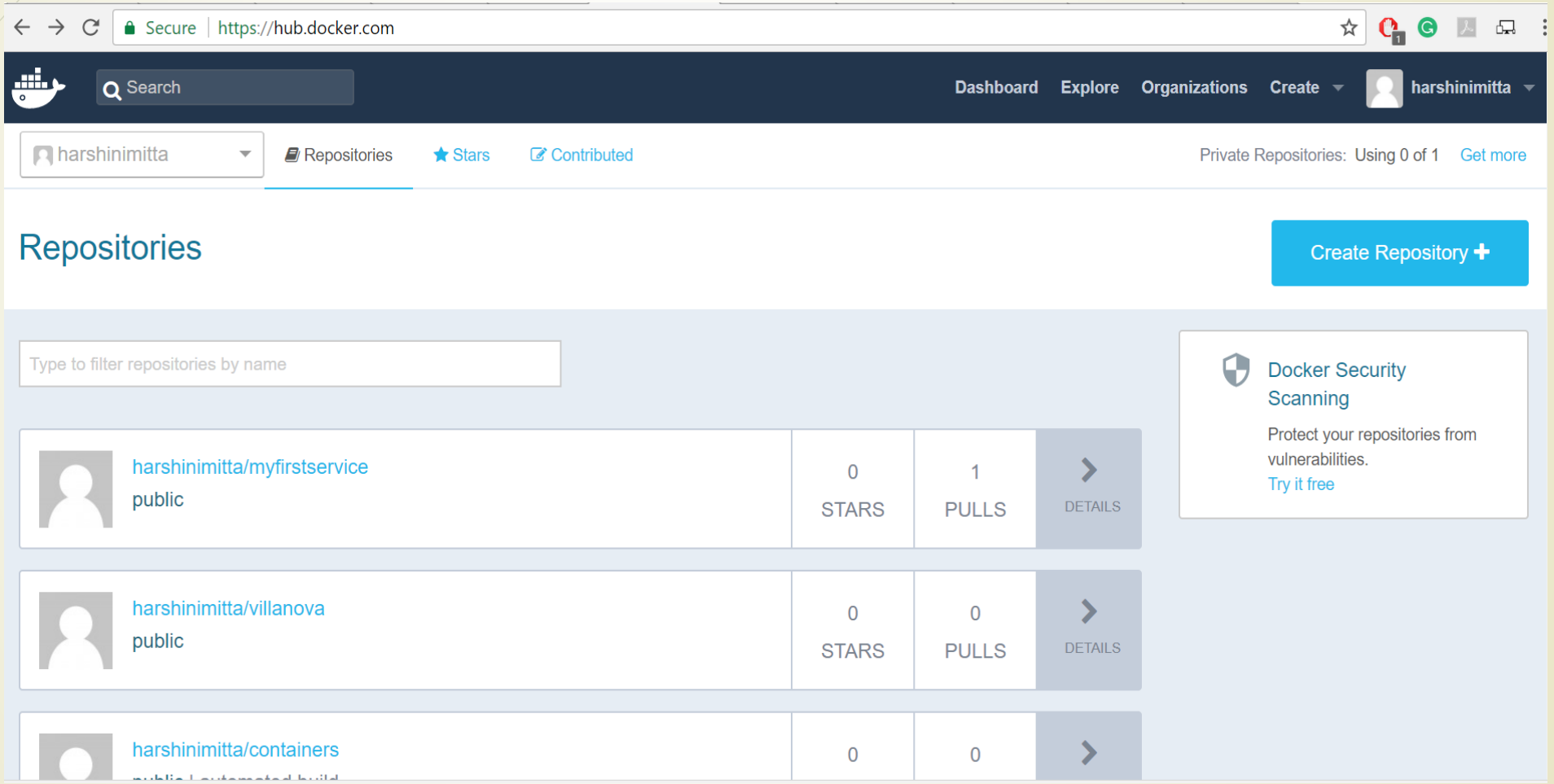


Container Image

Docker Hub Repo

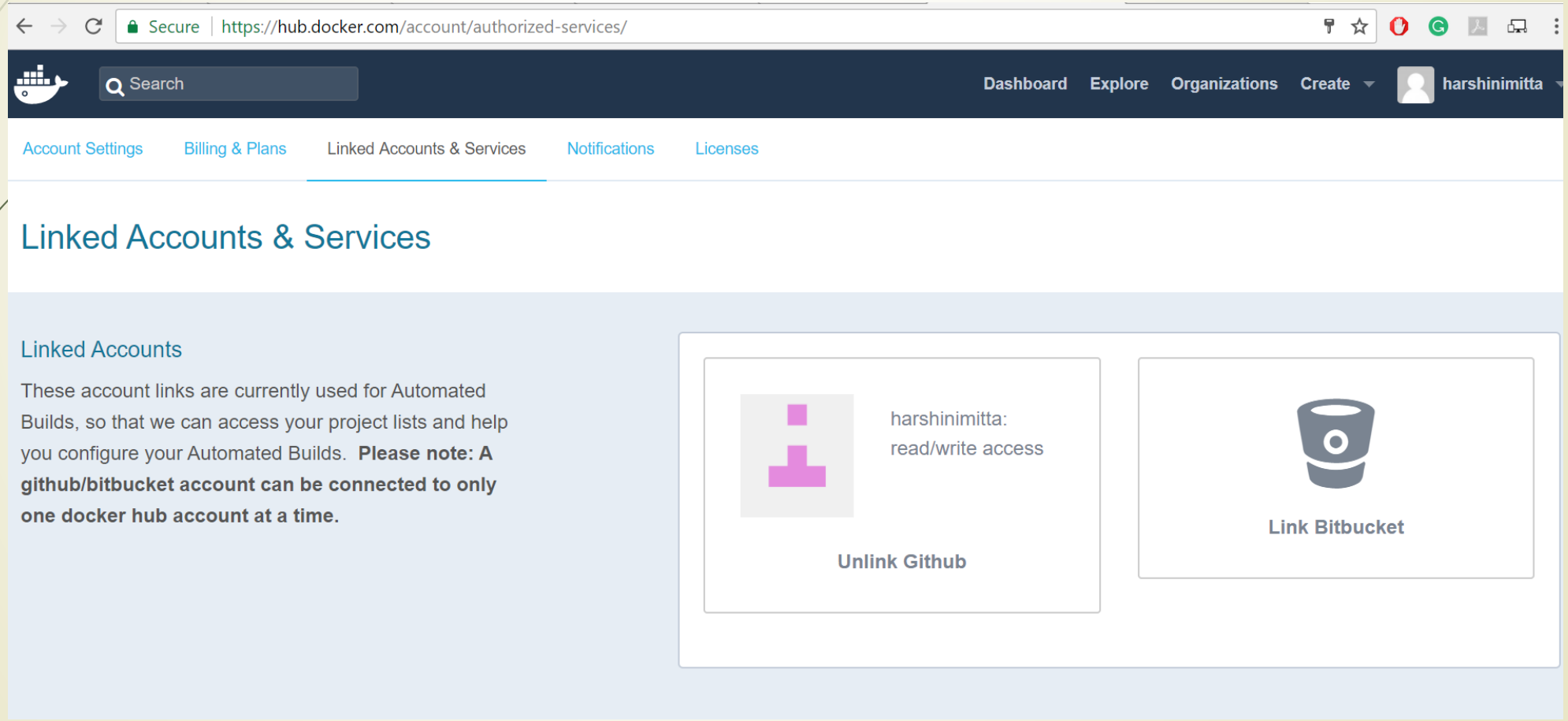
- Create a tag
  - $ docker tag <image_id> <tag_name>
  - $ docker tag db795886d47c harshinimitta/myfirstservice
- Login to Docker
  - $ export DOCKER_ID_USER=harshinimitta
  - $ docker login
    - Prompts you for username and password
- Push it to Repo
  - $ docker push harshinimitta/myfirstservice

Docker Hub Repo



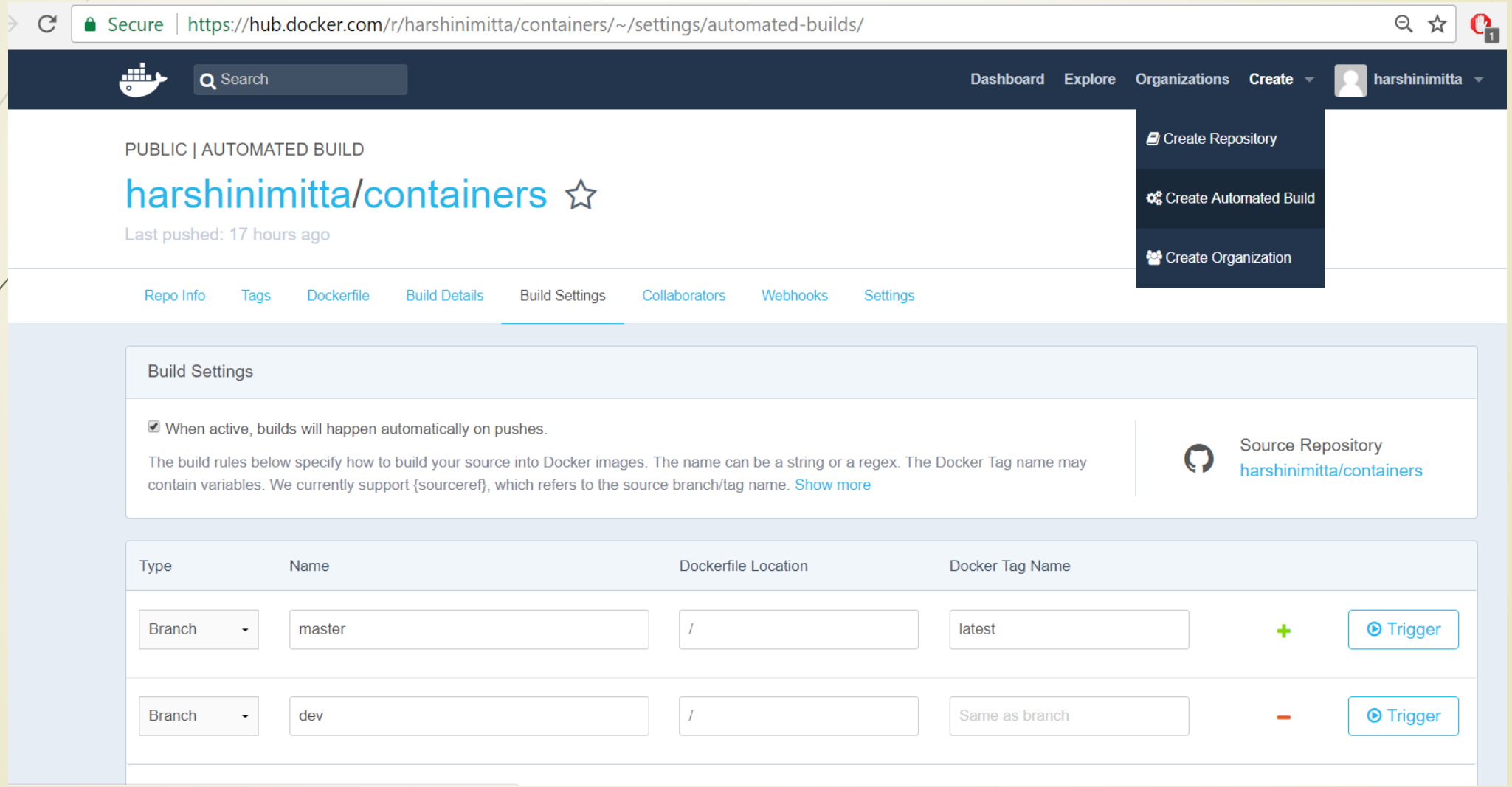
Automated Deployment

Navigate to **Profile > Settings > Linked Accounts & Services**

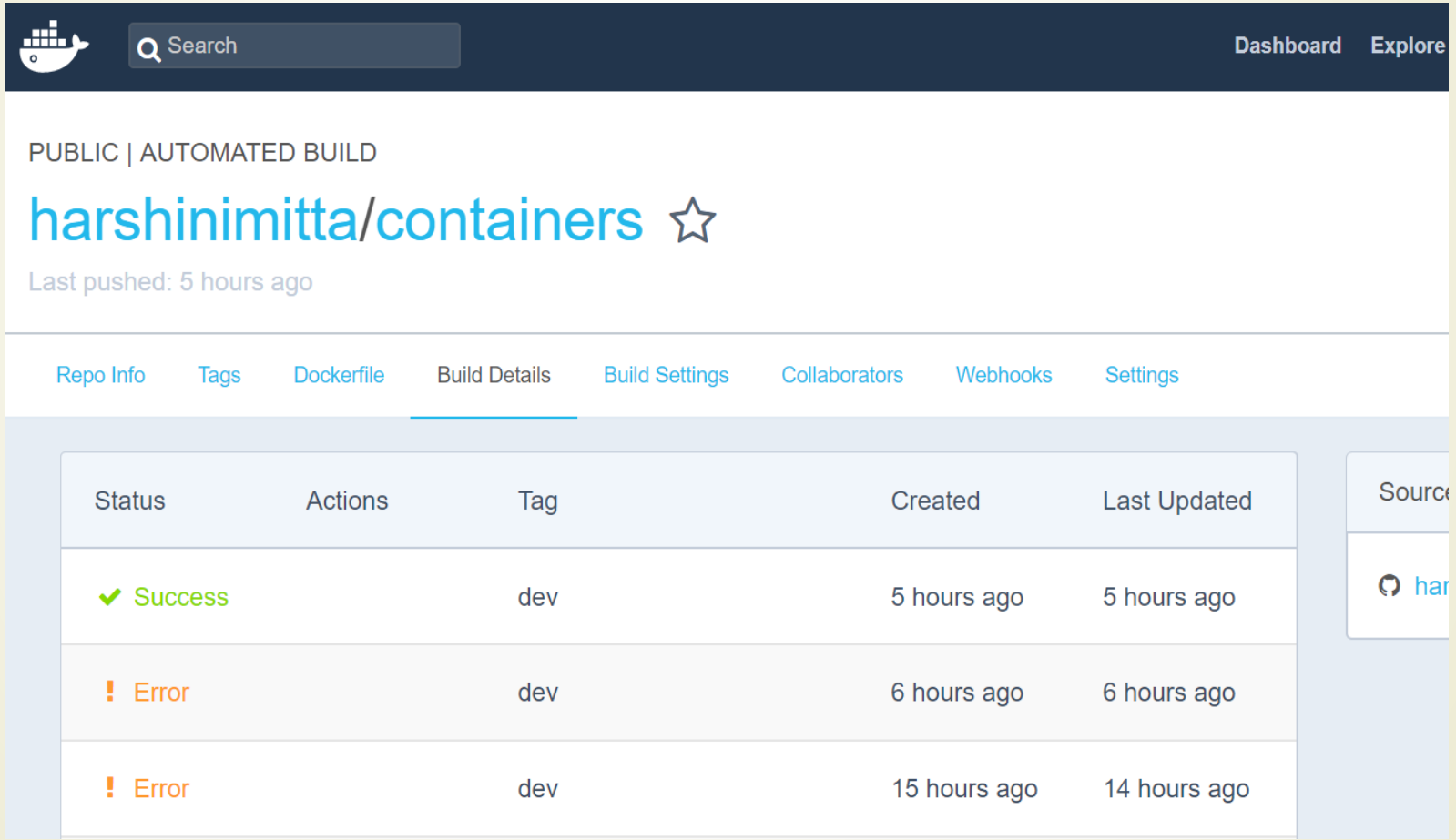


Create an automated build

Create > Create Automated Build



Build Details



Conclusion

- Expectation
- Best Thing about this course
- Challenges
- Expectations met?
- Suggestion

Thank You