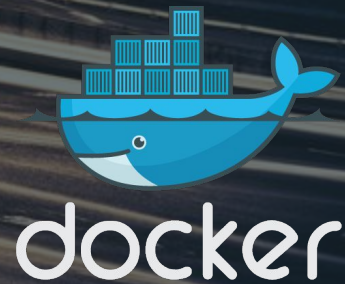


Docker in Continuous Integration



Module objectives

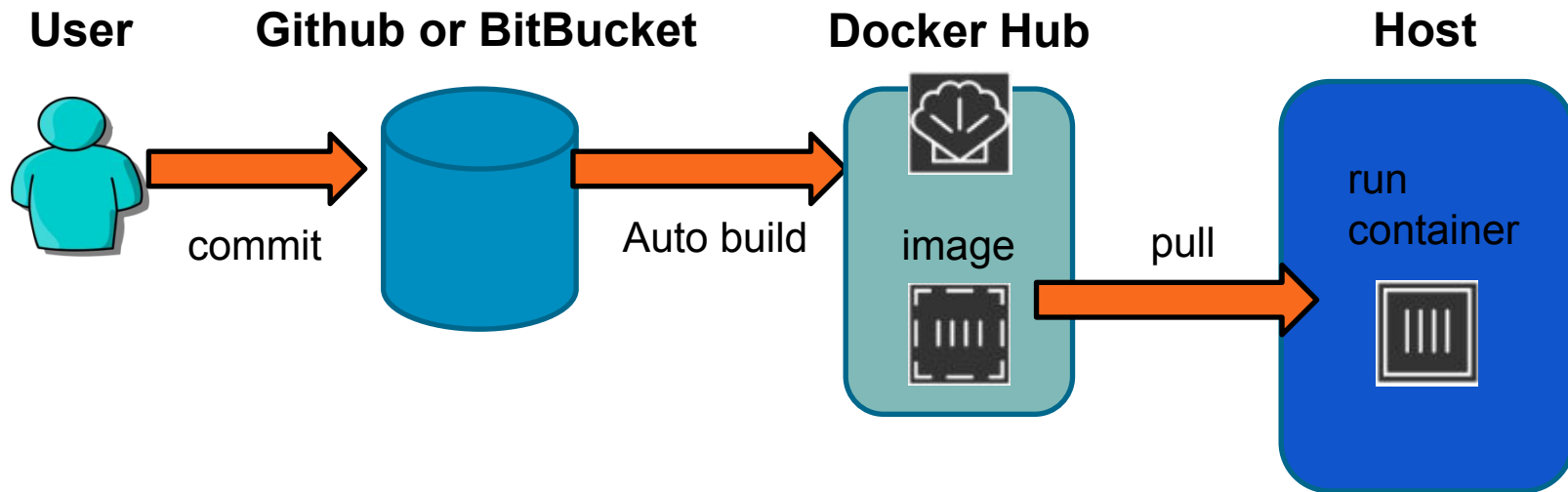
In this module we will:

- Explore ways we can fit Docker containers into our continuous integration process
- Setup an automated build in Docker Hub



Docker Hub Auto Build

- Docker Hub detects commits to source repository and builds the image
- Container is run during image build
- Testing done inside container



Setup an auto build example

- **Revision:** remember the simple “hello world” java application we built earlier?
- Let’s take that application and put it into a simple CI process using the Docker Hub auto build feature
- We will need to do the following:
 - Put our code into a repository (GitHub)
 - Setup an automated build on Docker Hub and have it connected to our GitHub account





Setup GitHub account


1. Go to <https://github.com/join> and setup a GitHub account. If you have an existing GitHub account, you can choose to use that one instead.

Join GitHub

The best way to design, build, and ship software.

 **Step 1:**
Set up a personal account

 **Step 2:**
Choose your plan

 **Step 3:**
Go to your dashboard

Create your personal account

Username

This will be your username — you can enter your organization's username next.

Email Address

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

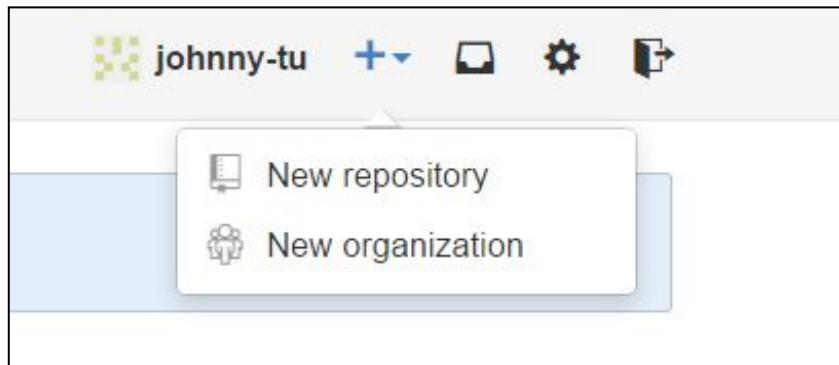
You'll love GitHub

- Unlimited** collaborators
- Unlimited** public repositories
- ✓ Great communication
- ✓ Friction-less development
- ✓ Open source community



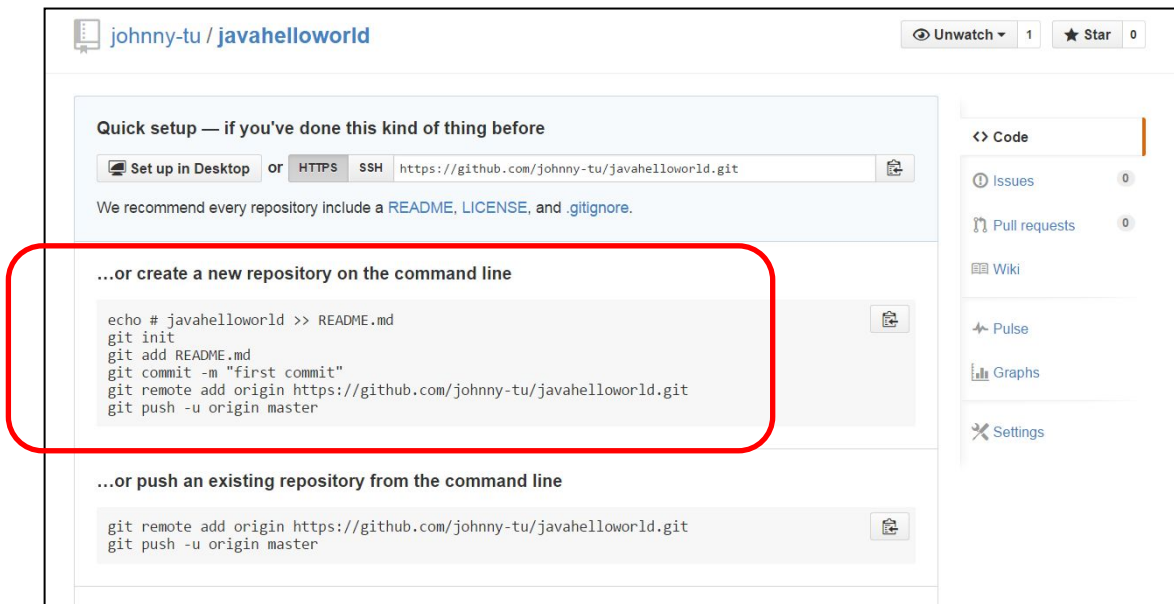
Create a new GitHub repository

- We will need a new repo for our application
- Repository name will be `javahelloworld`



Add existing code to GitHub repository

- Once you have created the repository, you will see a section on the page that contains step by step instruction on adding your existing code to the repository



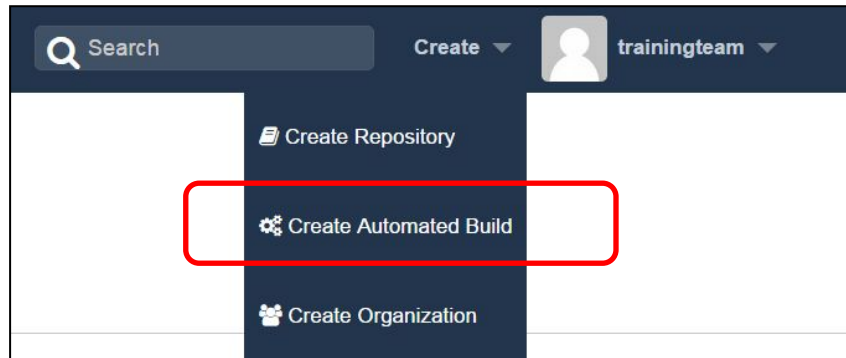
Push our code into the repository

- First initialise the git repository in our working directory
`git init`
- Then add the files we want to commit to the git staging area
`git add src/HelloWorld.java`
- Commit the code locally
`git commit -m "first commit"`
- Add our GitHub repository as a remote repository
`git remote add origin https://github.com/<username>/<repo name>.git`
- Push our commit to the remote repository
`git push origin master`



Setup Docker Hub auto build

- Click “Create Automated Build”
- Link your Docker Hub account to GitHub or Bitbucket, if you have not configured this in your settings



You haven't linked to  GitHub or  Bitbucket yet.

[Link Accounts](#)



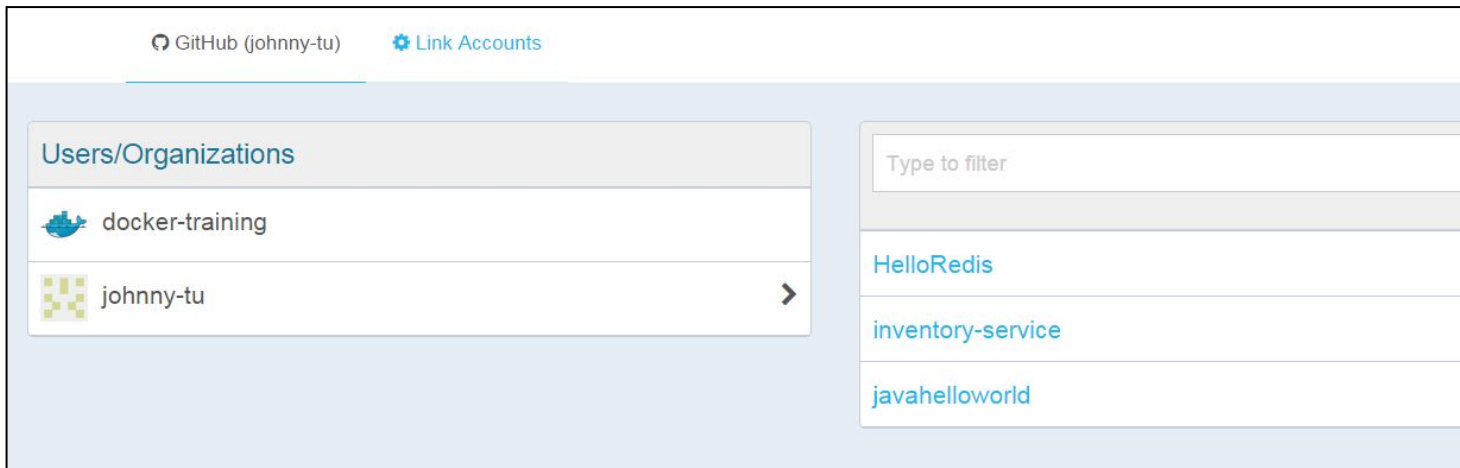
Select the repository provider

- Select GitHub or BitBucket and follow the screen prompts



Choose your GitHub repository

- Once linked, you will need to click “Create Automated Build” again
- If you have multiple GitHub accounts connected to DockerHub, they will appear on this screen
- Choose the right GitHub account and then choose the “javahelloworld” repository



Create the automated build

- By default the repository name of the automated build will be the same as the source code repository name.
- You can choose to run automated builds based on branches or tags
- The “Docker Tag Name” field specifies what tag newly built images are given

trainingteam

javahelloworld

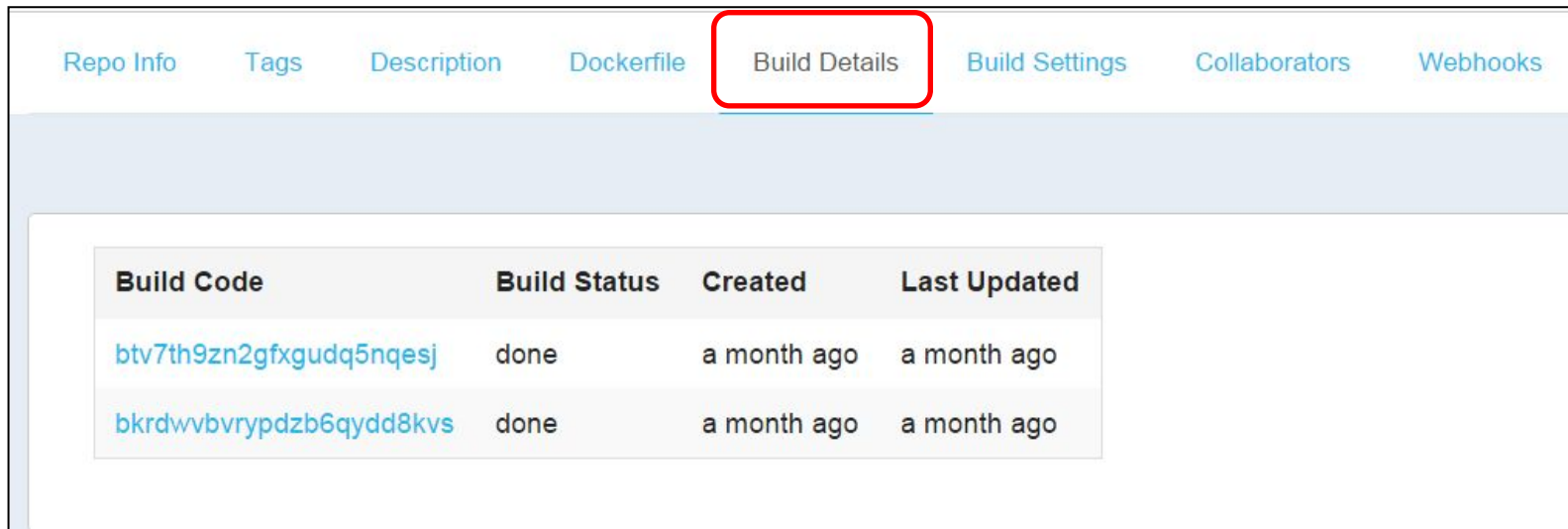
Short Description (100 Characters)

Type	Name	Dockerfile Location	Tag
Branch	master	/	latest



Checking progress and results

- An automated build repository in Docker Hub contains a “Build Details” tab
- The “Build Details” tab shows the history of the image being built

A screenshot of the Docker Hub interface showing the 'Build Details' tab. The tab is highlighted with a red rectangle. Below the navigation bar, there is a table with build history.

Repo Info	Tags	Description	Dockerfile	Build Details	Build Settings	Collaborators	Webhooks
Build Code	Build Status	Created	Last Updated				
btv7th9zn2gfxgudq5nqesj	done	a month ago	a month ago				
bkrdwvrvrypdzb6qydd8kvs	done	a month ago	a month ago				



Build log

- Click on the build id on the Build History tab to view the details of that build, including the build log

PUBLIC | AUTOMATED BUILD

trainingteam/javahelloworld-prod ☆

Last pushed: a month ago

[Repo Info](#) [Tags](#) [Description](#) [Dockerfile](#) [Build Details](#) [Build Settings](#) [Collaborators](#) [Webhooks](#) [Delete Repository](#)

Name	Value
error	
readme_contents	
created_at	a month ago
build_path	/
docker_tag	latest
source_url	https://github.com/johnny-tu/javahelloworld.git
build_code	btv7th9zn2gfgudq5nqesj
source_branch	master

[▶ Trigger a Build](#) [Source Project](#)

DOCKER PULL COMMAND

docker pull trainingteam/javahelloworld-prod

DESCRIPTION

This is our auto build

OWNER



Module summary

- There are many ways for Docker containers to fit into your continuous integration or continuous delivery process
- Docker Hub's auto build repository is one way for us to build and distribute production ready images

