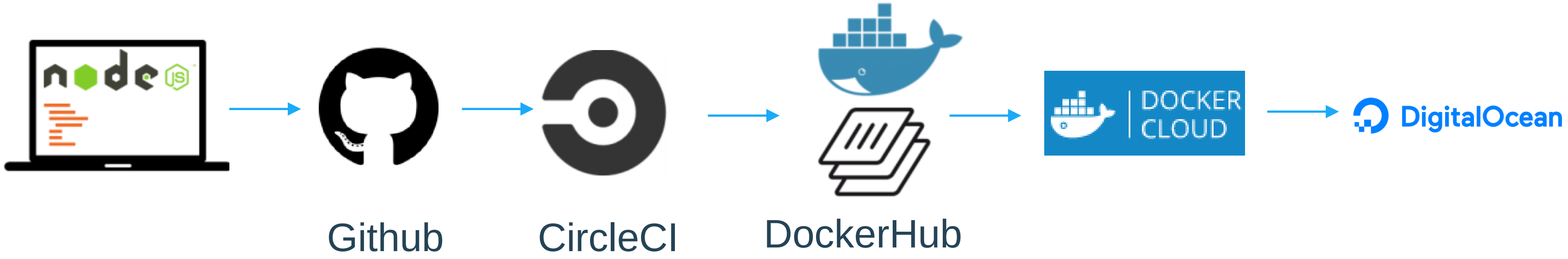
- Linas- 159.65.126.44
- Tadas - 159.65.114.171
- Vaidotas - 159.65.114.45
- Aurelijus - 159.65.126.115
- Lolita – 159.65.124.113
- Mindaugas – 159.89.99.102
- Deimantas - 159.65.118.52
- Rimantas – 159.65.118.204
- Nerijus - 159.65.118.189
- Ignas - 159.65.126.175

Docker in Continuous Integration

CI/CD workflow



App

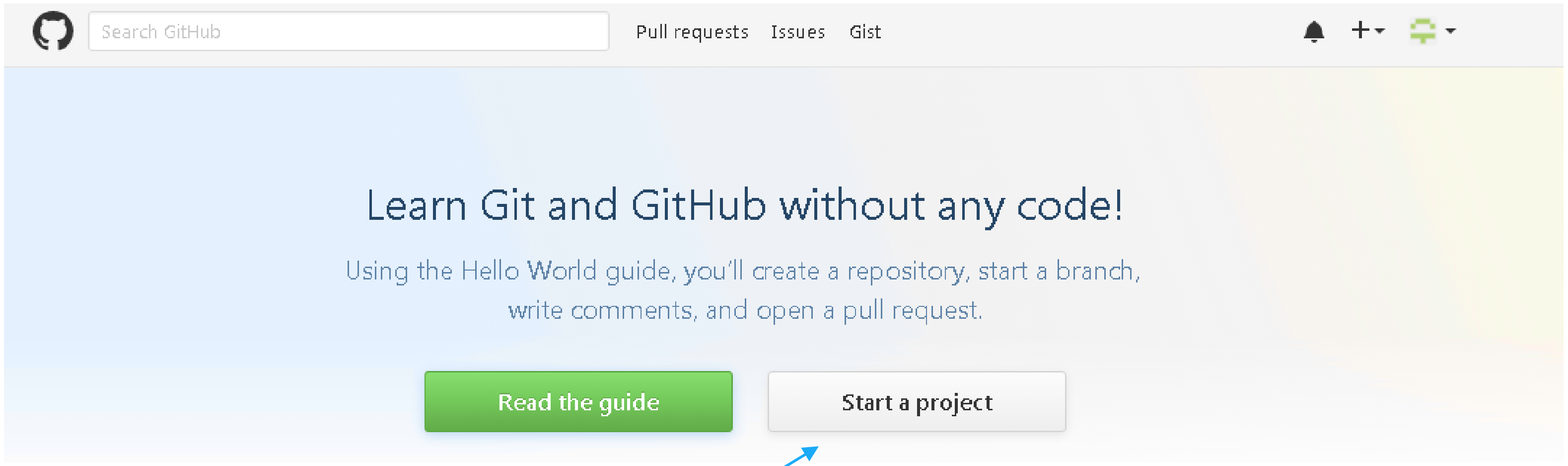
https://github.com/docker-4-devops/docker-ci

- Simple JS app
- Dockerfile to build image
- Mocha to test the App

Webhooks & Triggers

- Event-based triggers
- Crucial part of our CI/CI Workflow

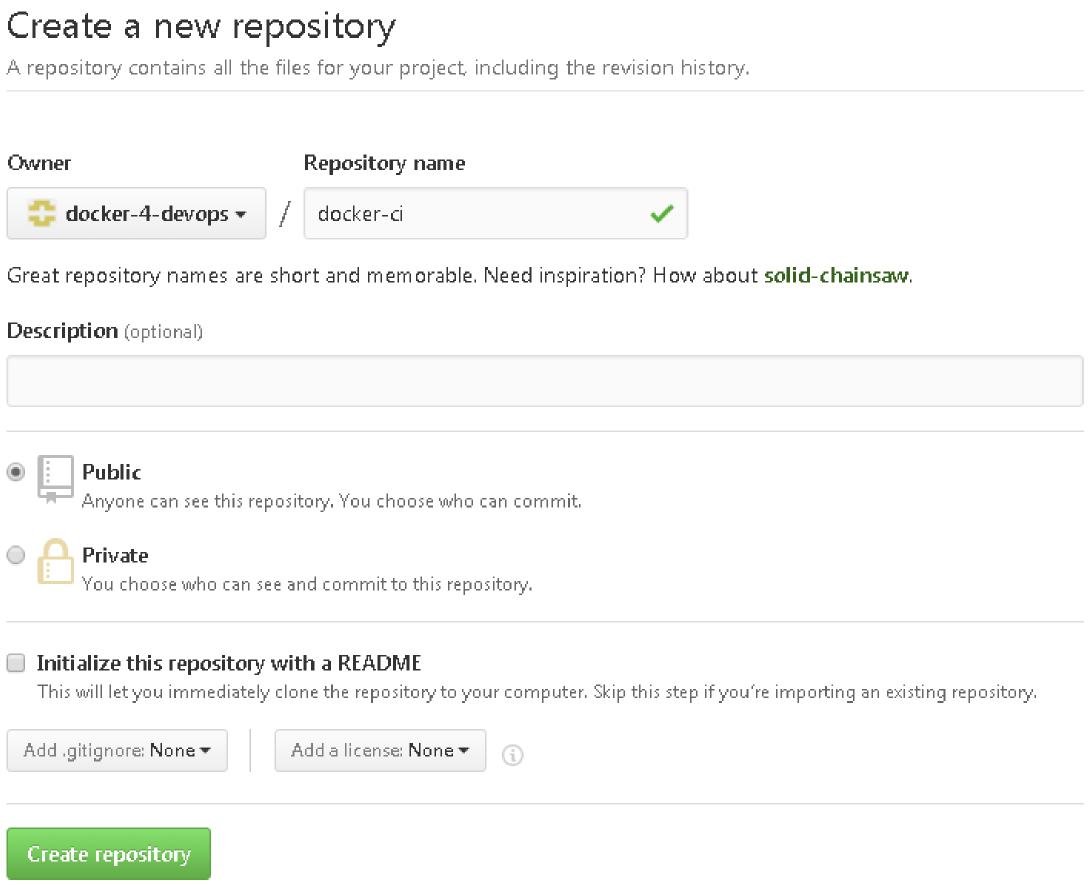
Start new project on github



Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Create repository



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner / **Repository name**

docker-4-devops / docker-ci ✓

Great repository names are short and memorable. Need inspiration? How about **solid-chainsaw**.

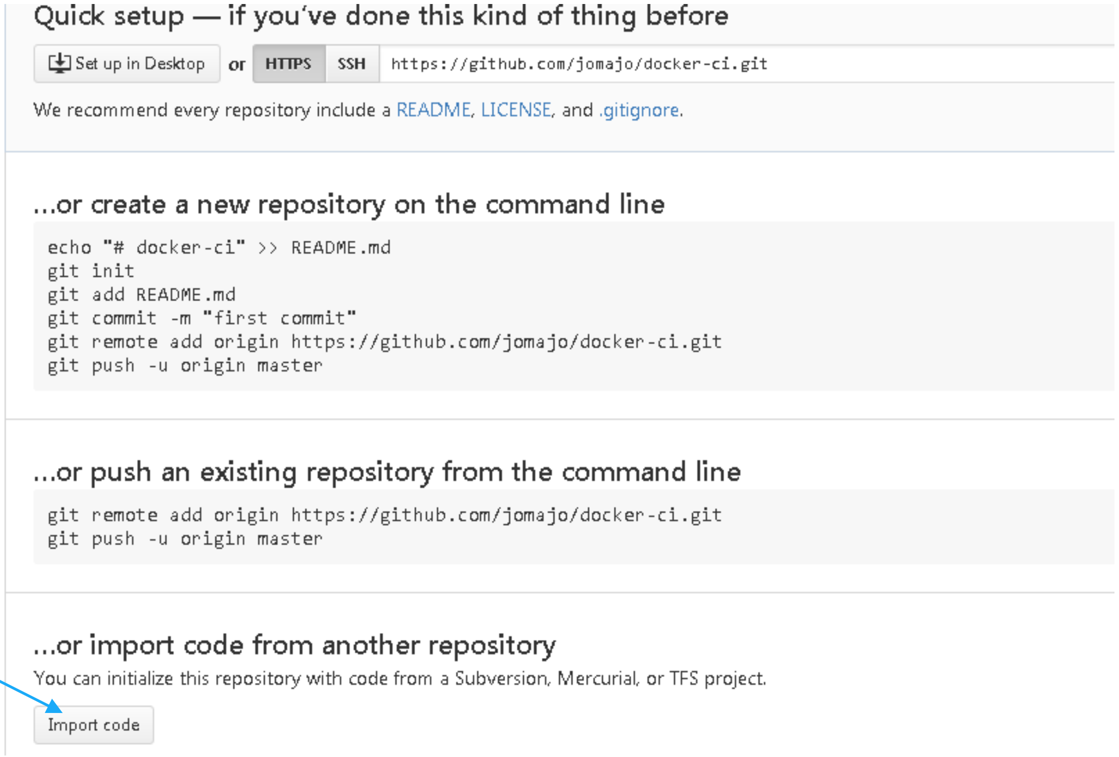
Description (optional)

● **Public**
Anyone can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Import code



Quick setup — if you've done this kind of thing before

https://github.com/jomajo/docker-ci.git

We recommend every repository include a README, LICENSE, and .gitignore.

…or create a new repository on the command line

```
echo "# docker-ci" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/jomajo/docker-ci.git
git push -u origin master
```

…or push an existing repository from the command line

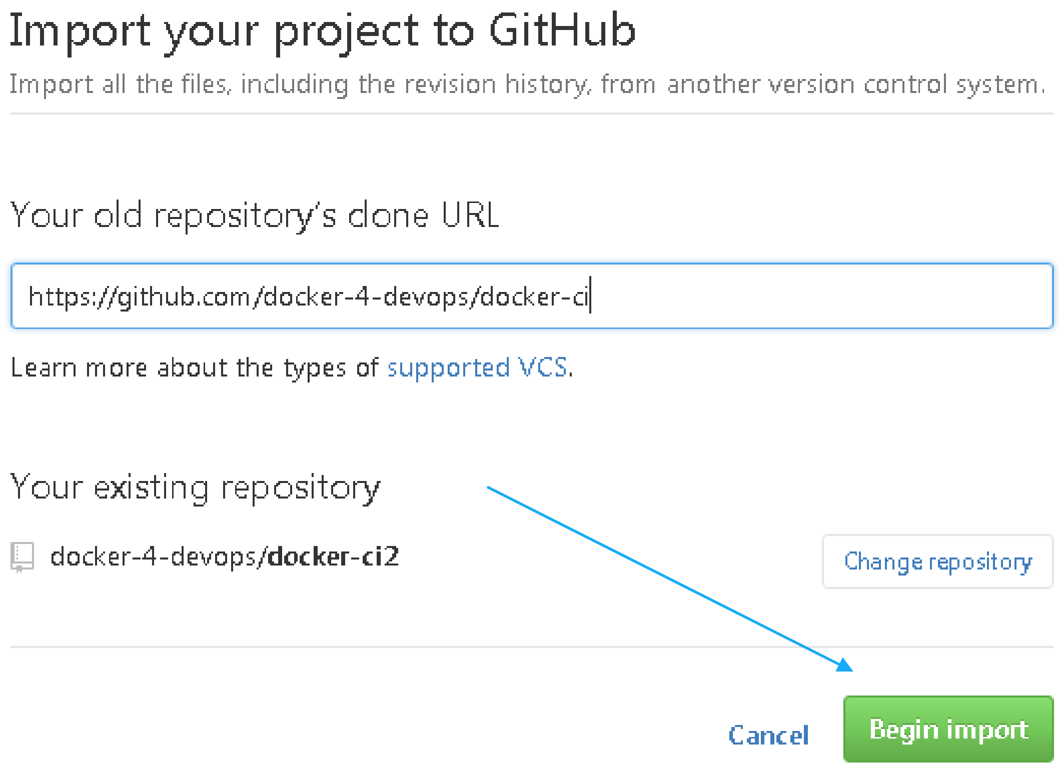
```
git remote add origin https://github.com/jomajo/docker-ci.git
git push -u origin master
```

…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import existing project

https://github.com/docker-4-devops/docker-ci



Import your project to GitHub

Import all the files, including the revision history, from another version control system.

Your old repository's clone URL

https://github.com/docker-4-devops/docker-ci

Learn more about the types of supported VCS.

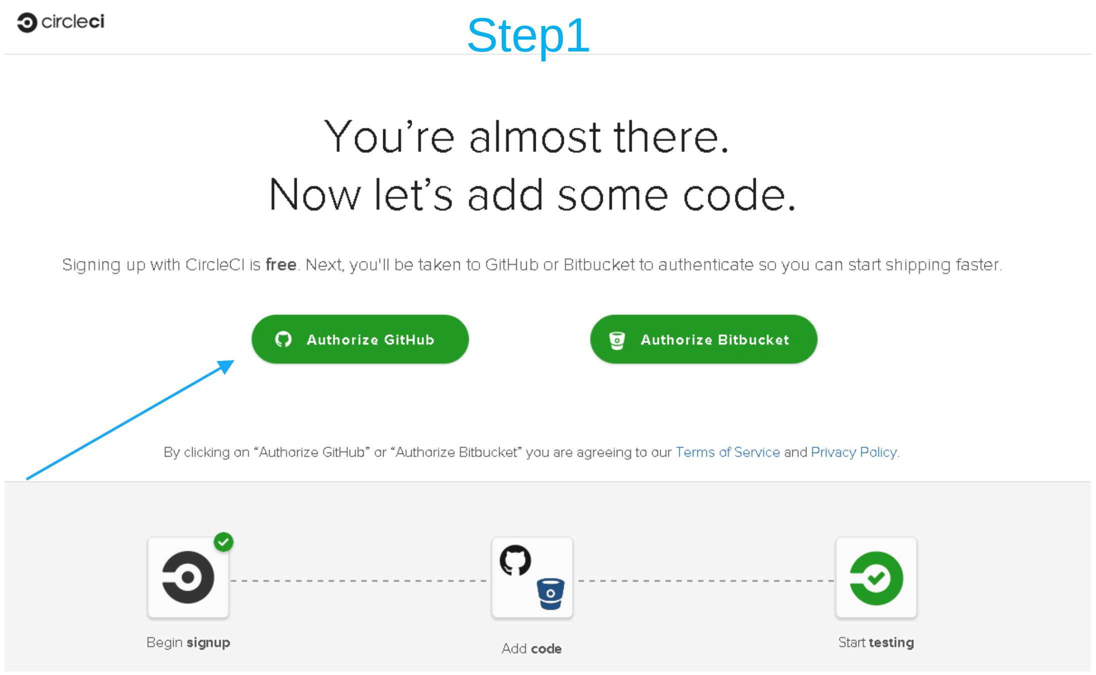
Your existing repository

docker-4-devops/**docker-ci2**

Sign up at CircleCi

https://circleci.com/signup/

Step1

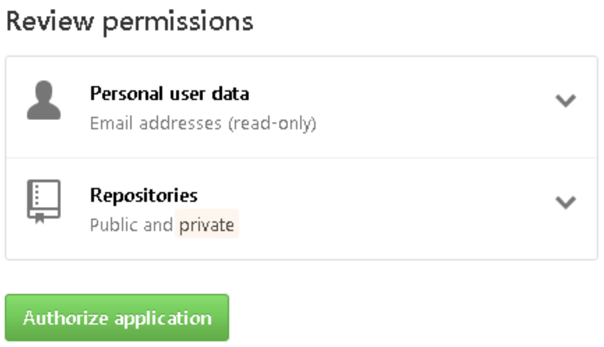

circleci

You're almost there.
Now let's add some code.

Signing up with CircleCI is **free**. Next, you'll be taken to GitHub or Bitbucket to authenticate so you can start shipping faster.

By clicking on "Authorize GitHub" or "Authorize Bitbucket" you are agreeing to our Terms of Service and Privacy Policy.

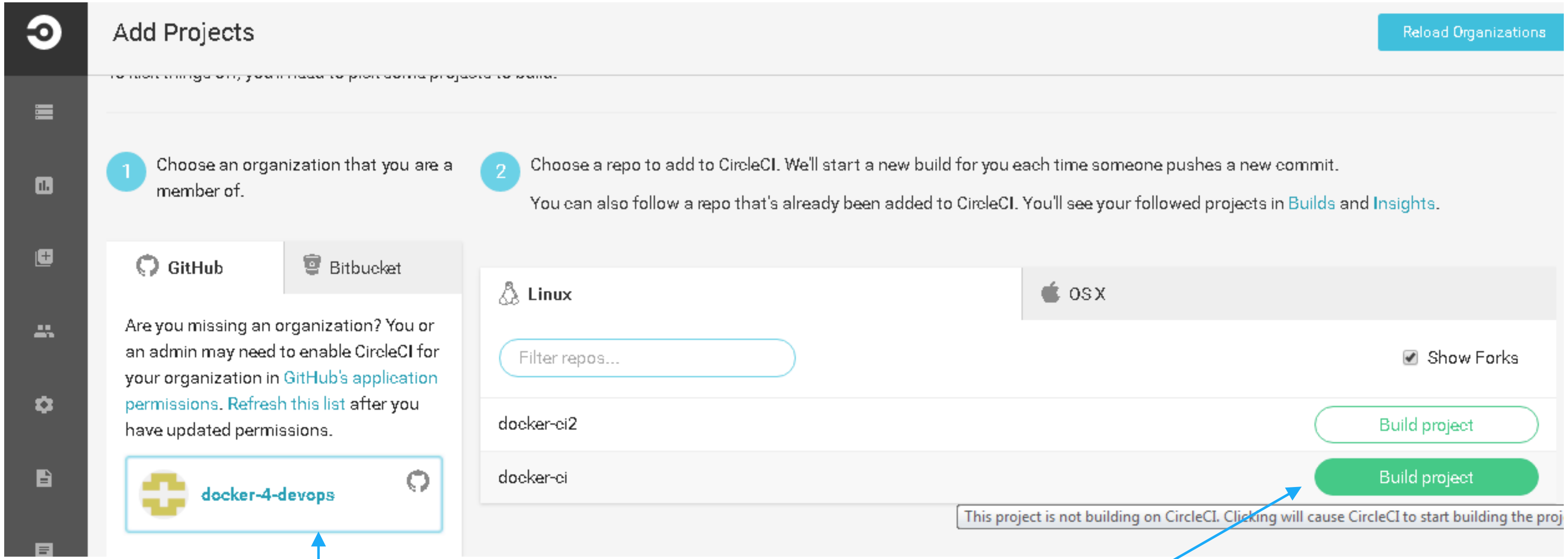
Step2


Review permissions

Personal user data
Email addresses (read-only)

Repositories
Public and private

Add project to Circle Ci



Add Projects

Reload Organizations

1 Choose an organization that you are a member of.

2 Choose a repo to add to CircleCI. We'll start a new build for you each time someone pushes a new commit.

You can also follow a repo that's already been added to CircleCI. You'll see your followed projects in Builds and Insights.

Are you missing an organization? You or an admin may need to enable CircleCI for your organization in GitHub's application permissions. Refresh this list after you have updated permissions.

docker-ci2 — Build project

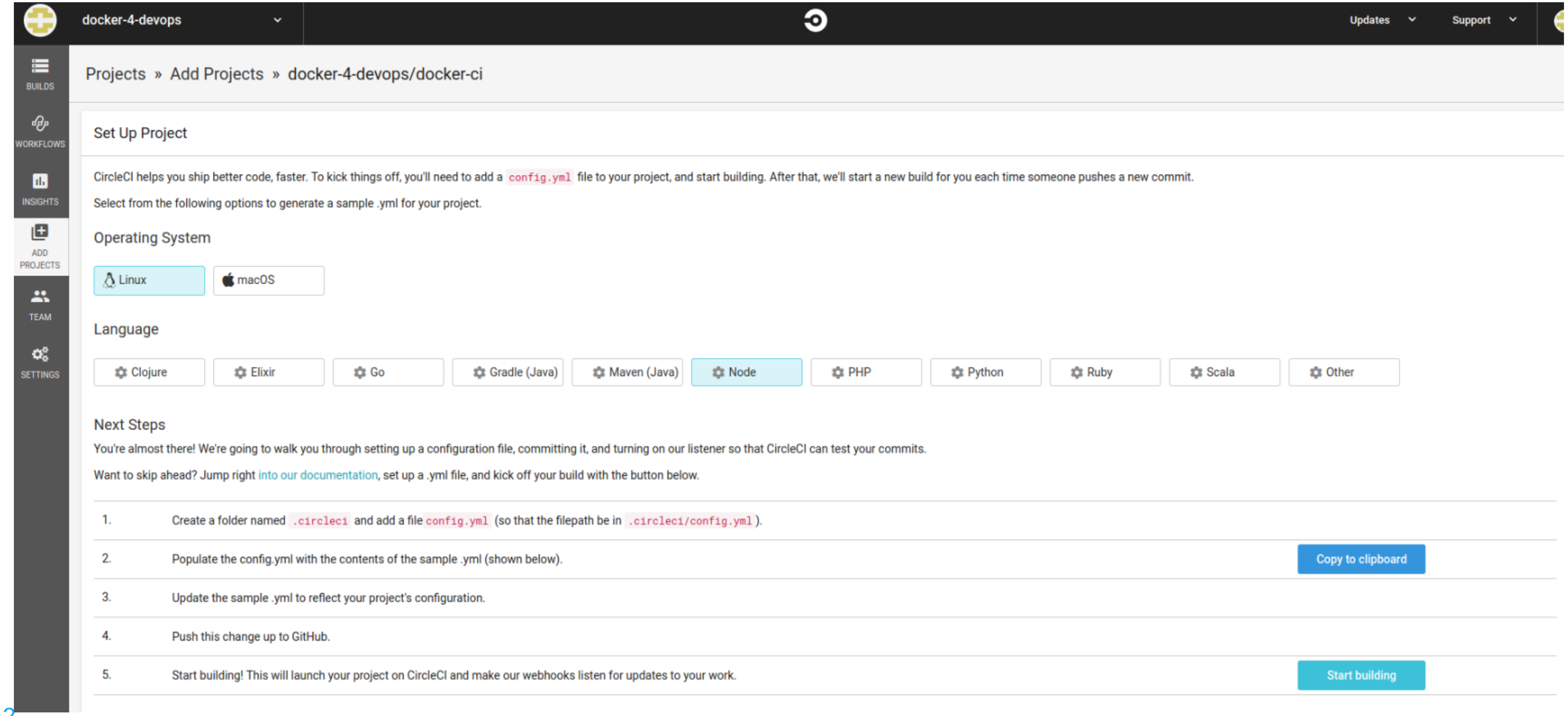
docker-ci — Build project

This project is not building on CircleCI. Clicking will cause CircleCI to start building the proj

Step1

Step2

Add project to Circle Ci



Projects » Add Projects » docker-4-devops/docker-ci

Set Up Project

CircleCI helps you ship better code, faster. To kick things off, you'll need to add a `config.yml` file to your project, and start building. After that, we'll start a new build for you each time someone pushes a new commit.

Select from the following options to generate a sample .yml for your project.

Operating System

[✓ Linux] macOS

Language

Clojure Elixir Go Gradle (Java) Maven (Java) [✓ Node] PHP Python Ruby Scala Other

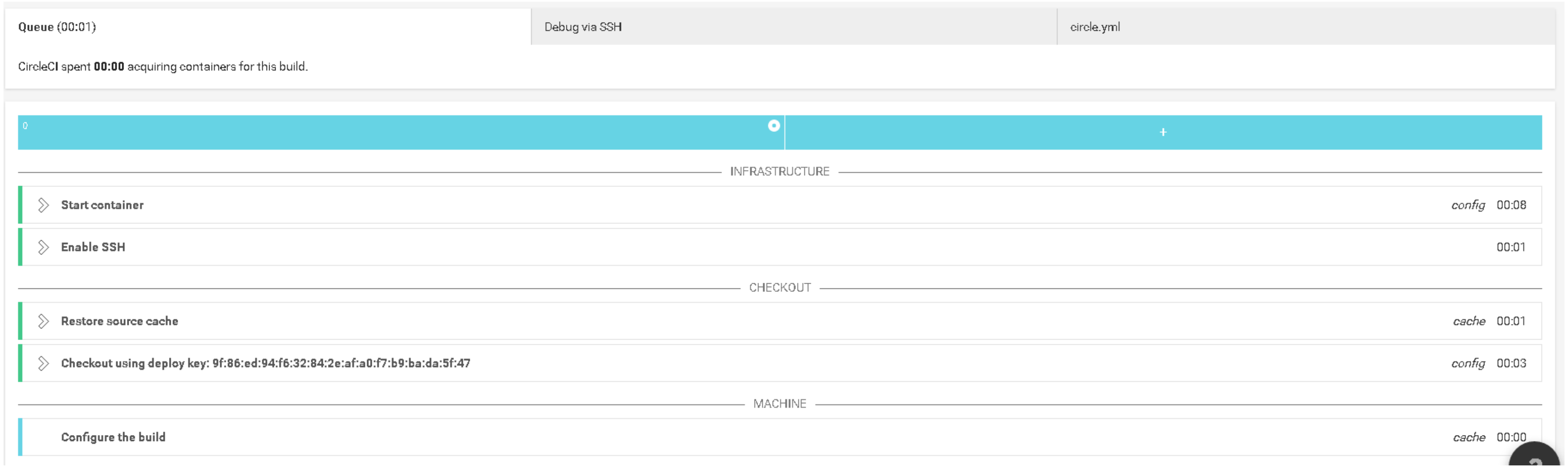
Next Steps

You're almost there! We're going to walk you through setting up a configuration file, committing it, and turning on our listener so that CircleCI can test your commits.

Want to skip ahead? Jump right into our documentation, set up a .yml file, and kick off your build with the button below.

1. Create a folder named `.circleci` and add a file `config.yml` (so that the filepath be in `.circleci/config.yml`).
2. Populate the config.yml with the contents of the sample .yml (shown below). Copy to clipboard
3. Update the sample .yml to reflect your project's configuration.
4. Push this change up to GitHub.
5. Start building! This will launch your project on CircleCI and make our webhooks listen for updates to your work. Start building

Build starts immediately!



Queue (00:01)

CircleCI spent **00:00** acquiring containers for this build.

INFRASTRUCTURE

Start container	*config*	00:08
Enable SSH		00:01

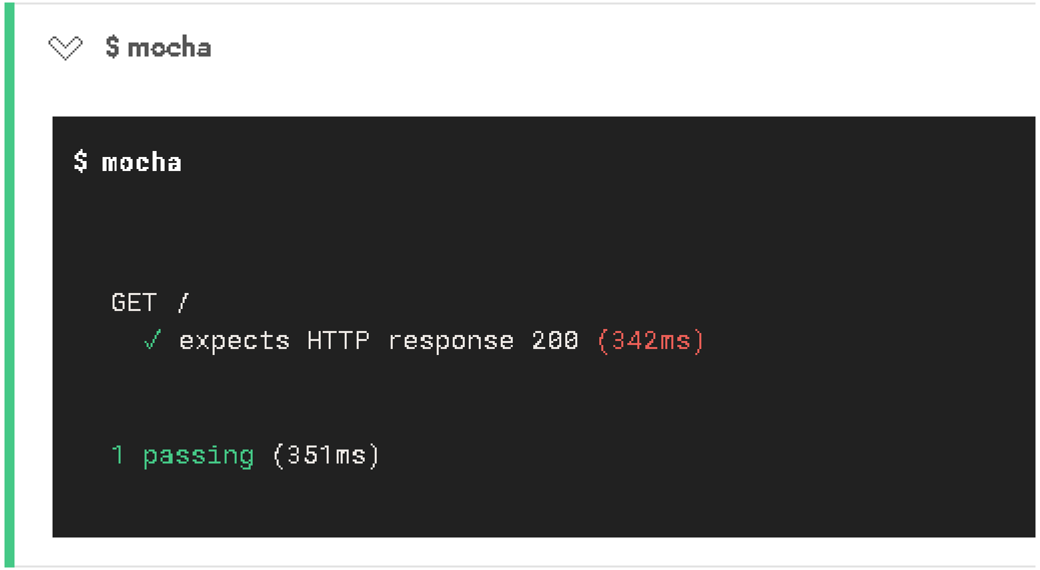
CHECKOUT

Restore source cache	*cache*	00:01
Checkout using deploy key: 9f:86:ed:94:f6:32:84:2e:af:a0:f7:b9:ba:da:5f:47	*config*	00:03

MACHINE

Configure the build	*cache*	00:00

Check if tests are passing!



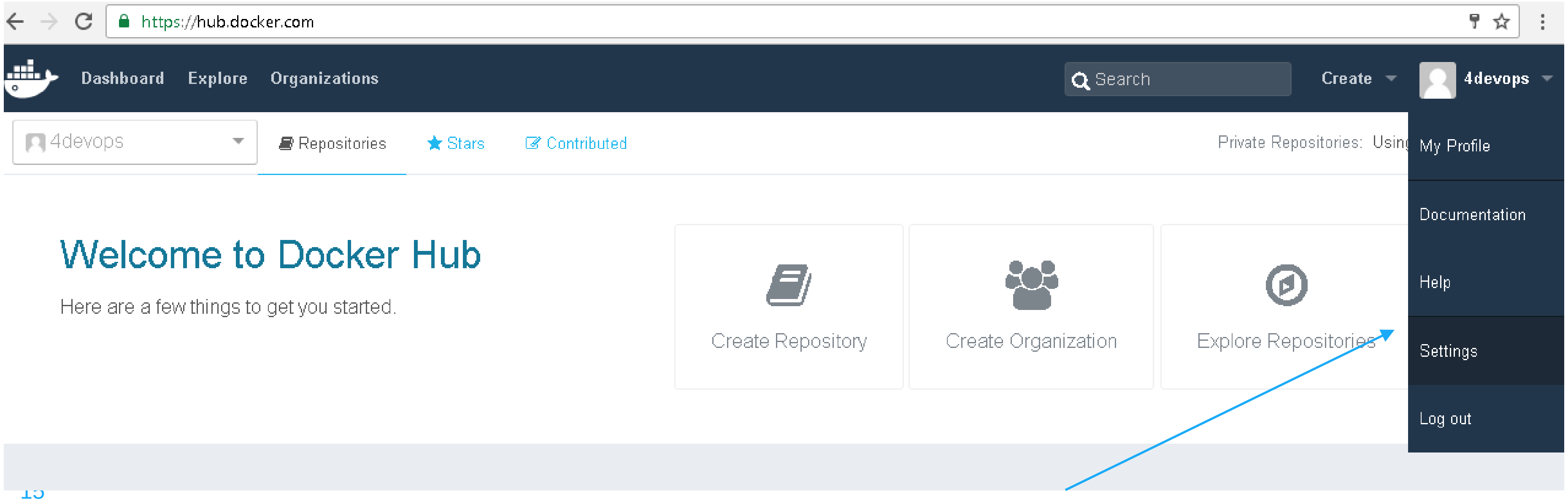
$ mocha

```
$ mocha



  GET /
    ✓ expects HTTP response 200 (342ms)


  1 passing (351ms)
```

Next-step - Dockerhub

https://hub.docker.com/

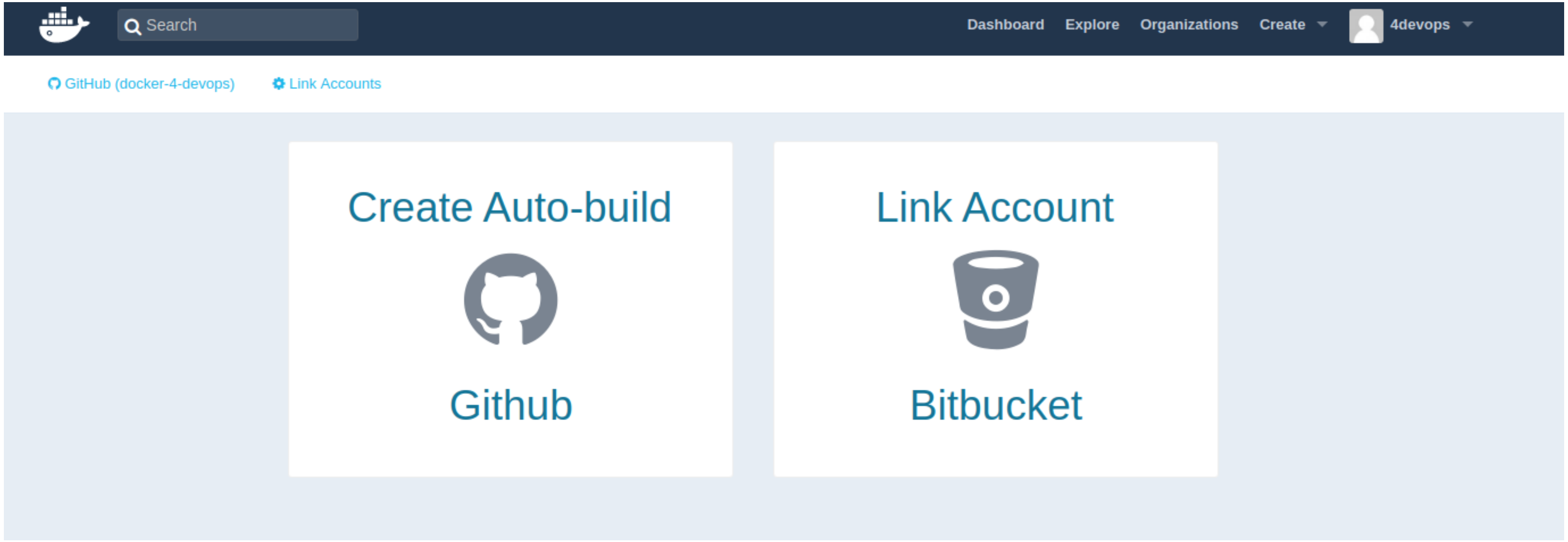
- Link Github account by going to:
  - Settings -> Linked Accounts & Services



Welcome to Docker Hub

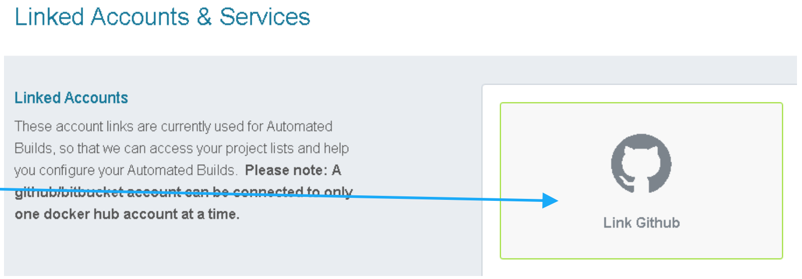
Here are a few things to get you started.

Link Your github account



Link Github account to DockerHub

Step1

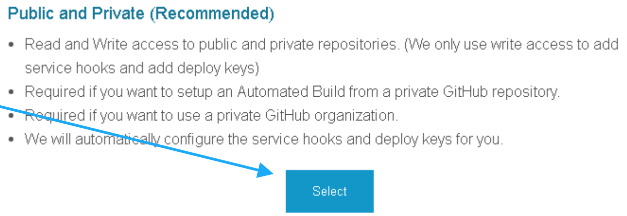


Linked Accounts & Services

Linked Accounts

These account links are currently used for Automated Builds, so that we can access your project lists and help you configure your Automated Builds. **Please note: A github/bitbucket account can be connected to only one docker hub account at a time.**

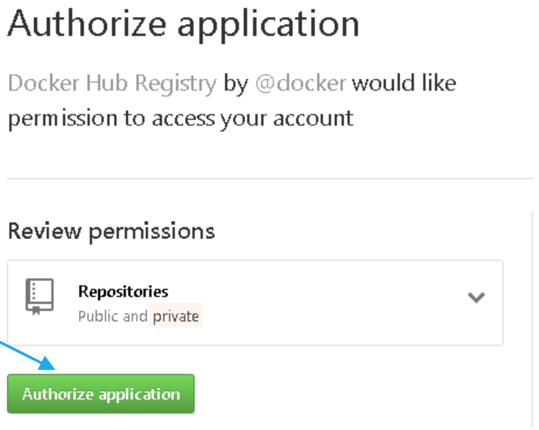
Step2



Public and Private (Recommended)

- Read and Write access to public and private repositories. (We only use write access to add service hooks and add deploy keys)
- Required if you want to setup an Automated Build from a private GitHub repository.
- Required if you want to use a private GitHub organization.
- We will automatically configure the service hooks and deploy keys for you.

Step3



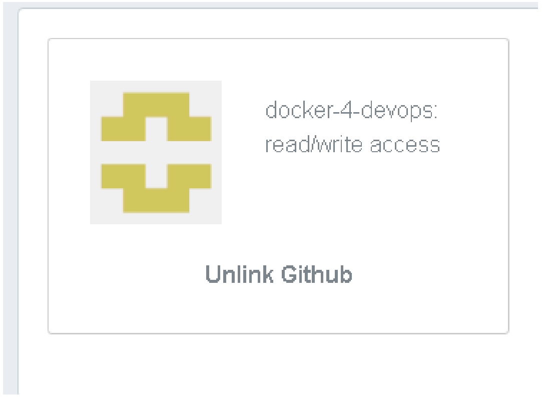
Authorize application

Docker Hub Registry by @docker would like permission to access your account

Review permissions

Repositories
Public and private

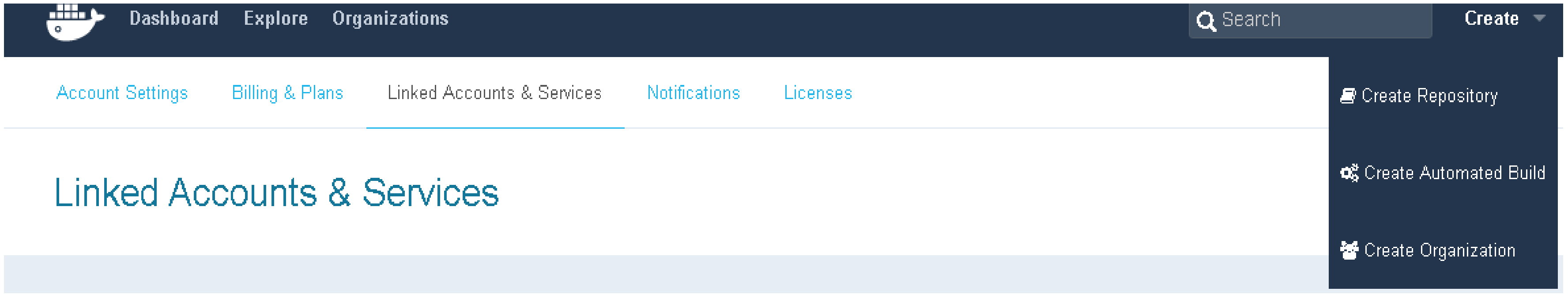
Step4



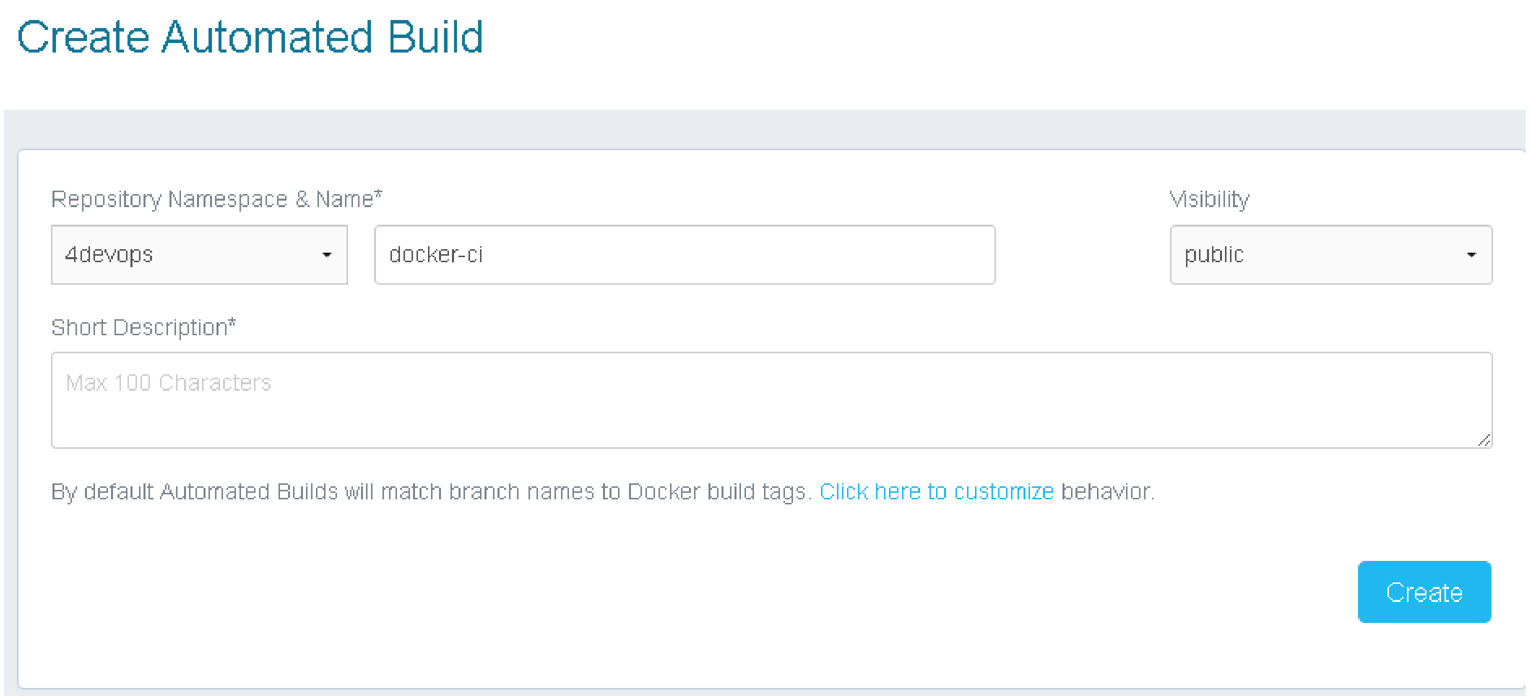
docker-4-devops: read/write access

Create automated build repository on DockerHub

- Create automated build repository



Create automated build repository on DockerHub


Create Automated Build

Repository Namespace & Name*

4devops

docker-ci

Visibility

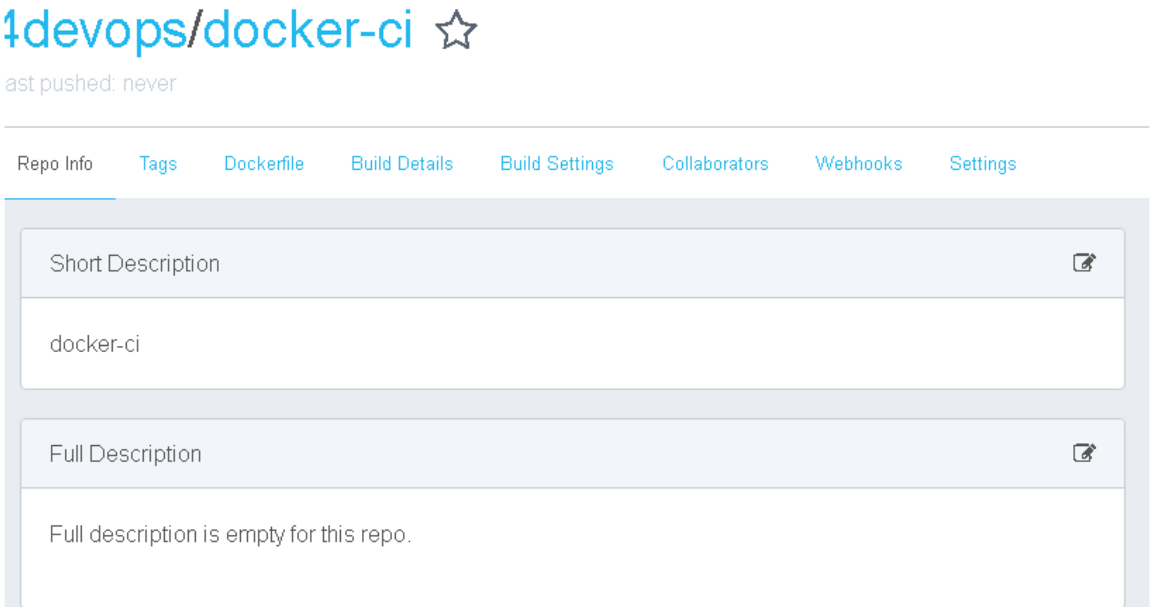
public

Short Description*

Max 100 Characters

By default Automated Builds will match branch names to Docker build tags. Click here to customize behavior.

Automated Build Repository



4devops/docker-ci

ast pushed: never

Short Description

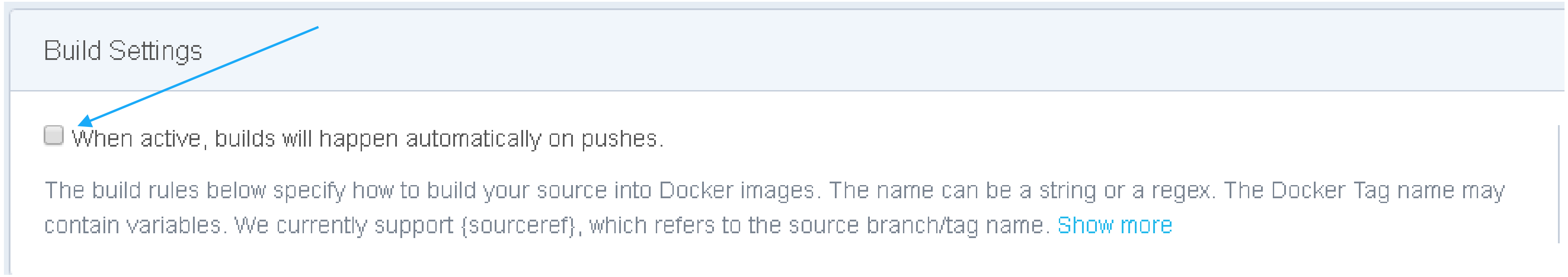
docker-ci

Full Description

Full description is empty for this repo.

Uncheck the option below under “Build Settings”

We want to build image only if tests are passing!

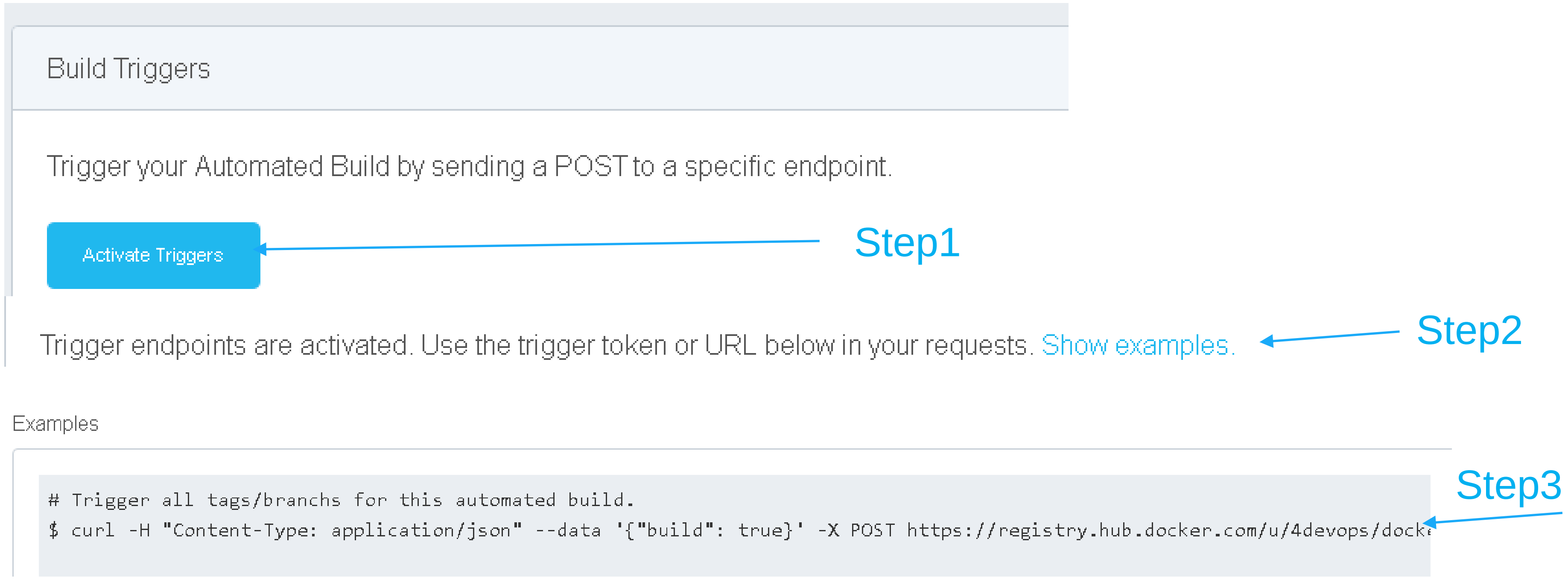


Build Settings

☐ When active, builds will happen automatically on pushes.

The build rules below specify how to build your source into Docker images. The name can be a string or a regex. The Docker Tag name may contain variables. We currently support {sourceref}, which refers to the source branch/tag name. Show more

Activate Triggers under “Build Settings”



Build Triggers

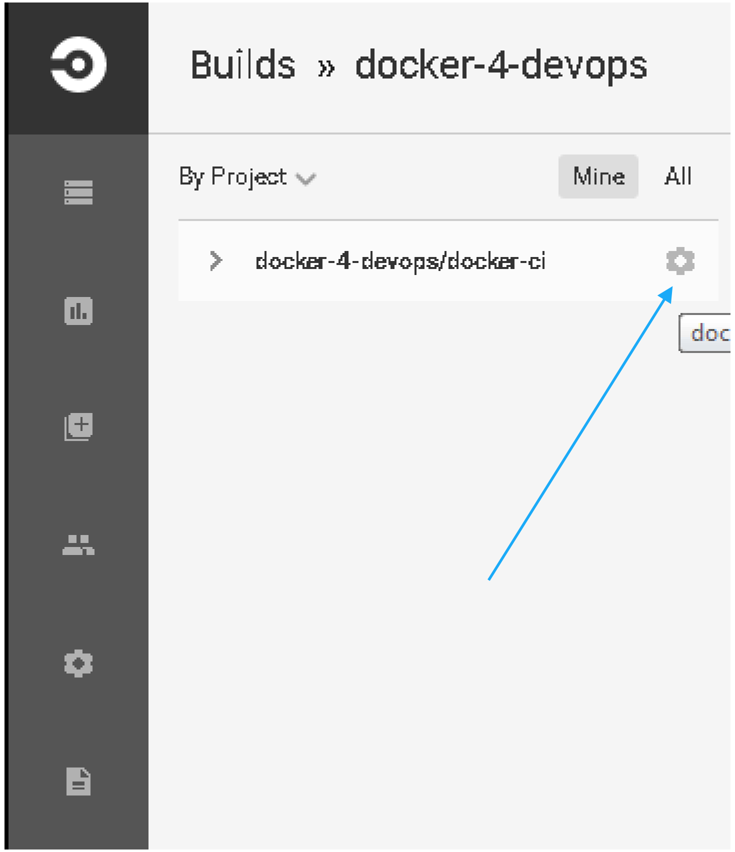
Trigger your Automated Build by sending a POST to a specific endpoint.

Trigger endpoints are activated. Use the trigger token or URL below in your requests. Show examples.

Examples

```
# Trigger all tags/branchs for this automated build.
$ curl -H "Content-Type: application/json" --data '{"build": true}' -X POST https://registry.hub.docker.com/u/4devops/dock
```

Copy Trigger URL to Circle CI

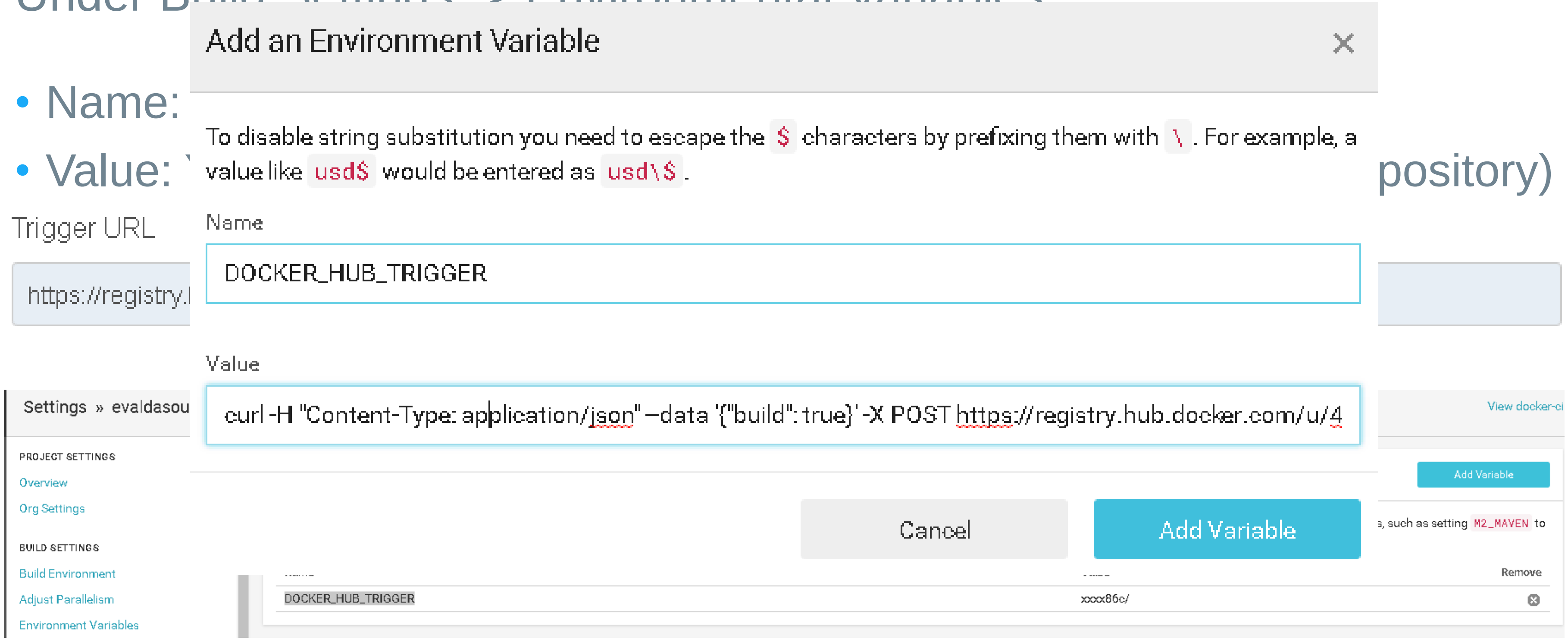


Builds » docker-4-devops

Add Trigger as Environment Variable

Under Build Settings -> Environmental Variables

- Name:
- Value: ` … pository)



Trigger URL

https://registry.l

Settings » evaldasou

PROJECT SETTINGS

Overview

Org Settings

BUILD SETTINGS

Build Environment

Adjust Parallelism

Environment Variables

Add an Environment Variable

To disable string substitution you need to escape the `$` characters by prefixing them with `\`. For example, a value like `usd$` would be entered as `usd\$`.

Name

DOCKER_HUB_TRIGGER

Value

```
curl -H "Content-Type: application/json" –data '{"build": true}' -X POST https://registry.hub.docker.com/u/4
```

Cancel | Add Variable

View docker-ci

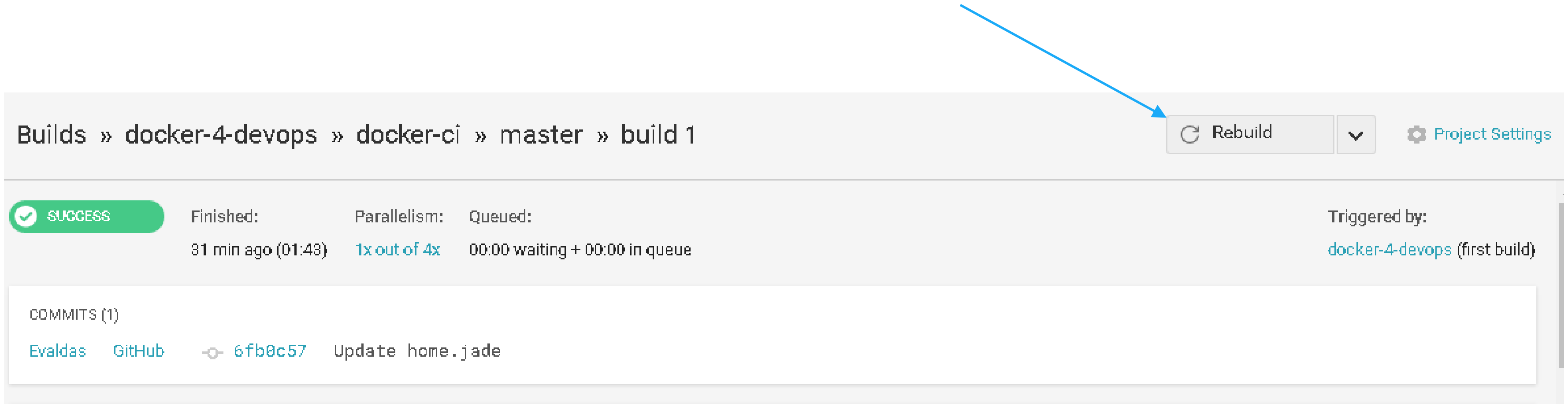
Add Variable

s, such as setting `M2_MAVEN` to

Name	Value	Remove
DOCKER_HUB_TRIGGER	xxxx86c/	⊗

Rebuild code under CircleCi

To Trigger Dockerfile build on DockerHub



Builds » docker-4-devops » docker-ci » master » build 1

Finished: 31 min ago (01:43)

Parallelism: 1x out of 4x

Queued: 00:00 waiting + 00:00 in queue

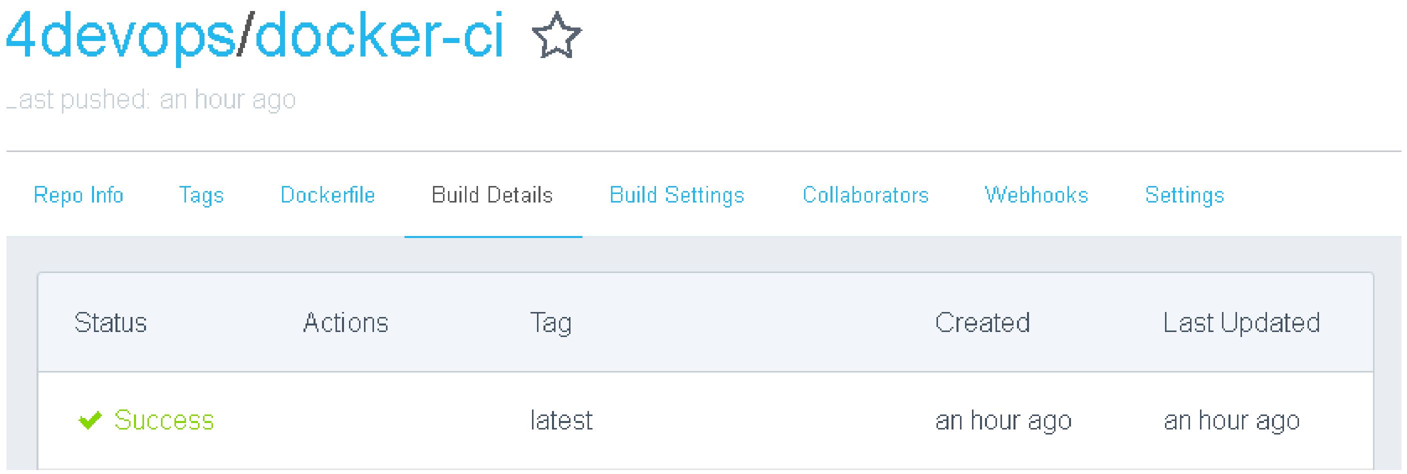
Triggered by: docker-4-devops (first build)

COMMITS (1)

Evaldas GitHub 6fb0c57 Update home.jade

Do we have new docker image on DockerHub?

- Check if app tests have passed on Circle CI
- Image was build and saved to Dockerhub
  (image build from Dockerfile (which is stored on Github)



4devops/docker-ci ☆

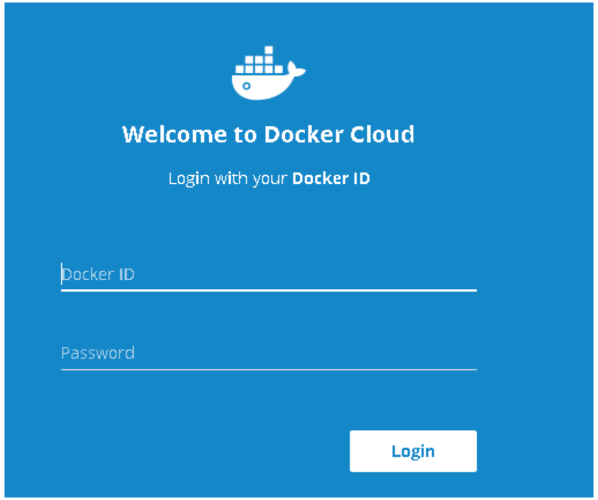
_ast pushed: an hour ago

Status	Actions	Tag	Created	Last Updated
✓ Success		latest	an hour ago	an hour ago

Login to DockerCloud

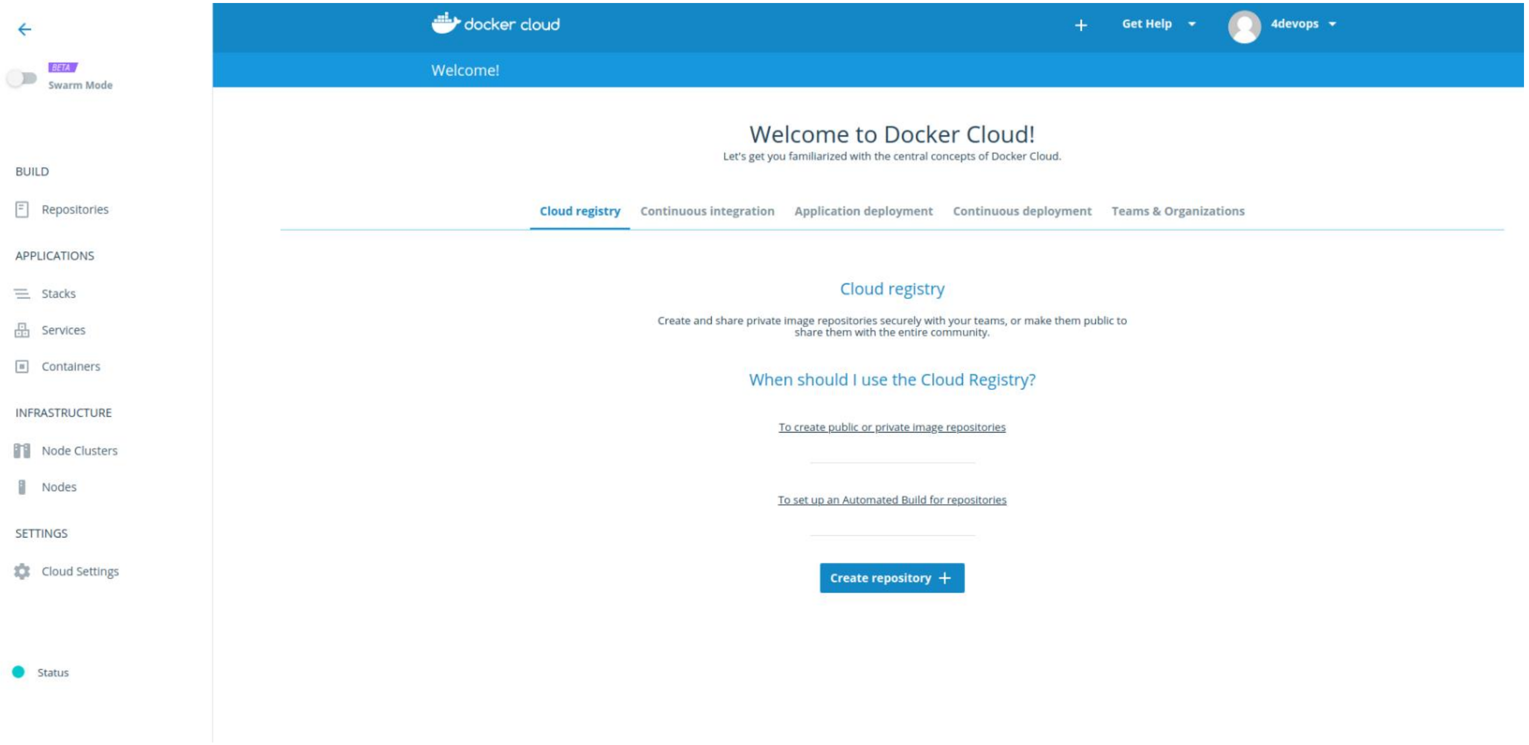
https://cloud.docker.com/

- For DockerCloud you can use same account as for Dockerhub
  - Choose login with Docker ID option.


Welcome to Docker Cloud

Login with your **Docker ID**

Docker Cloud



Welcome to Docker Cloud!

Let's get you familiarized with the central concepts of Docker Cloud.

Cloud registry

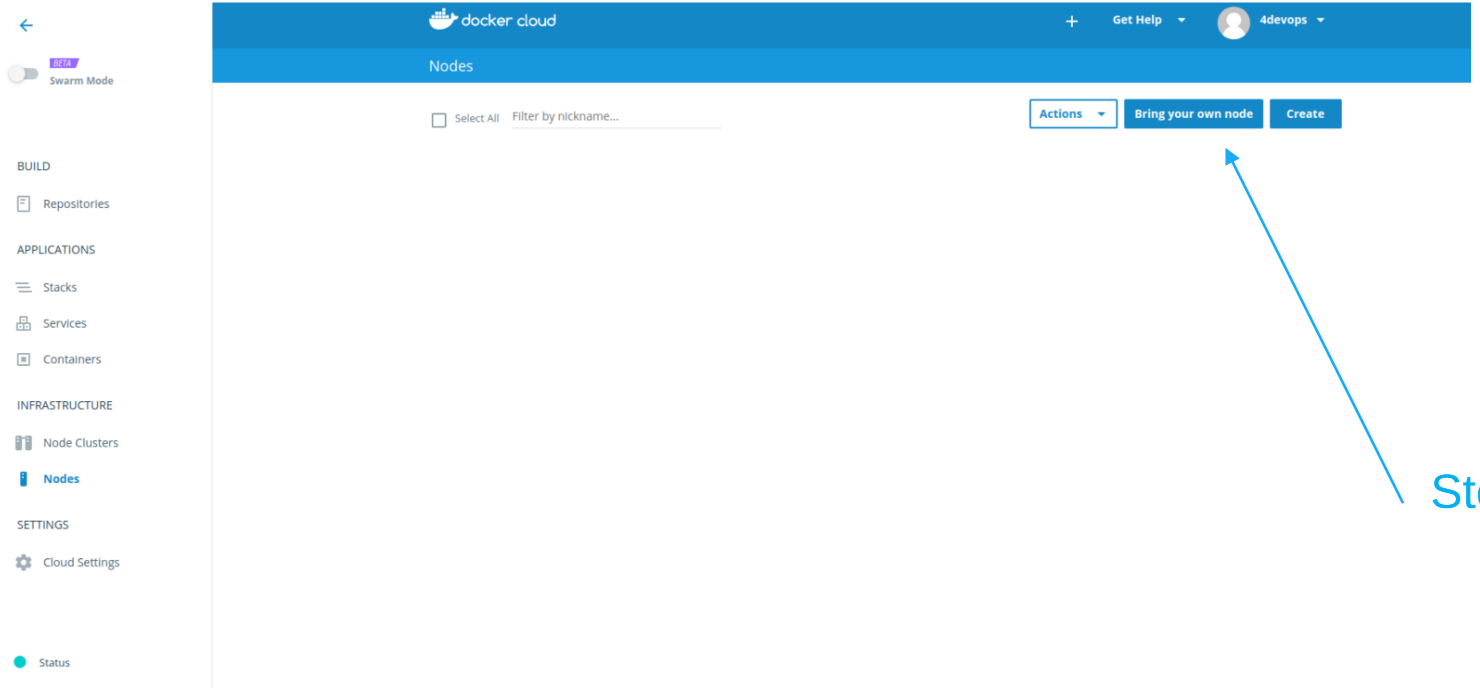
Create and share private image repositories securely with your teams, or make them public to share them with the entire community.

When should I use the Cloud Registry?

To create public or private image repositories

To set up an Automated Build for repositories

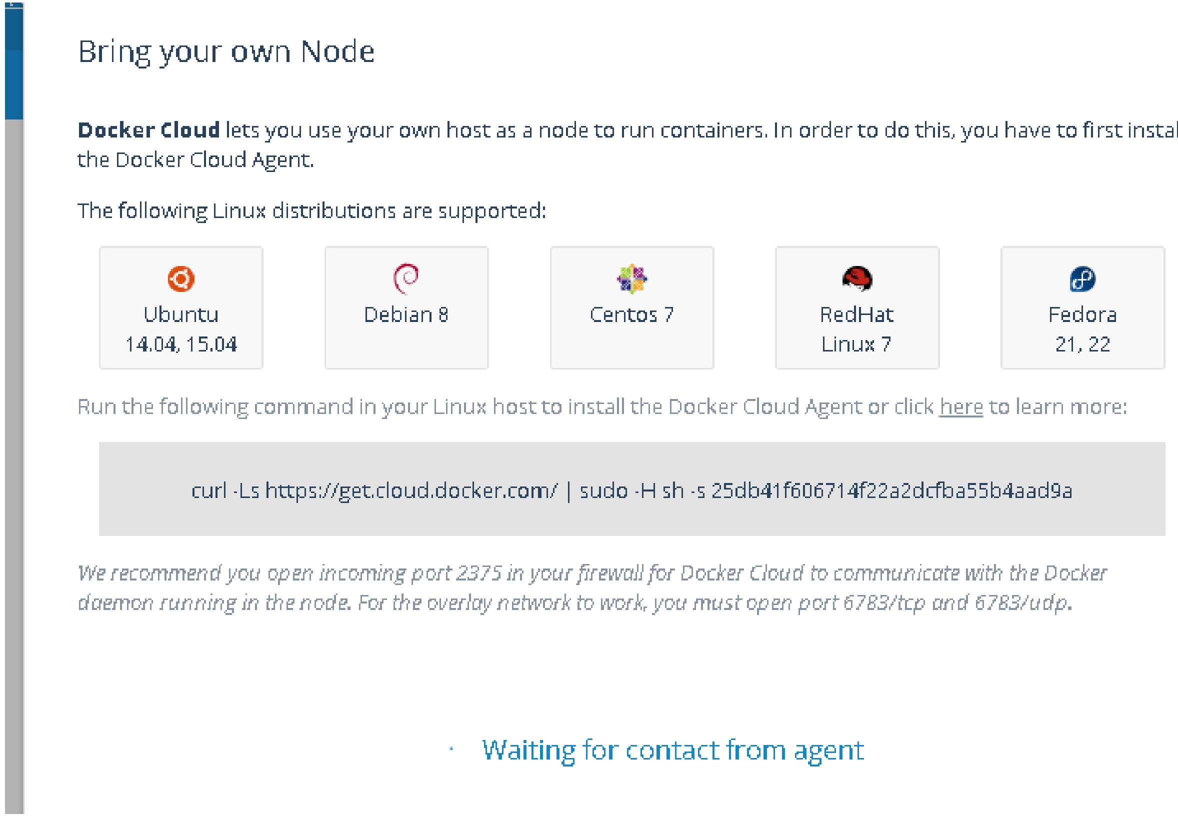
Create a node on Docker Cloud



Step1

Step2

Bring your own Node



Bring your own Node

Docker Cloud lets you use your own host as a node to run containers. In order to do this, you have to first install the Docker Cloud Agent.

The following Linux distributions are supported:

Run the following command in your Linux host to install the Docker Cloud Agent or click here to learn more:

```
curl -Ls https://get.cloud.docker.com/ | sudo -H sh -s 25db41f606714f22a2dcfba55b4aad9a
```

We recommend you open incoming port 2375 in your firewall for Docker Cloud to communicate with the Docker daemon running in the node. For the overlay network to work, you must open port 6783/tcp and 6783/udp.

Waiting for contact from agent

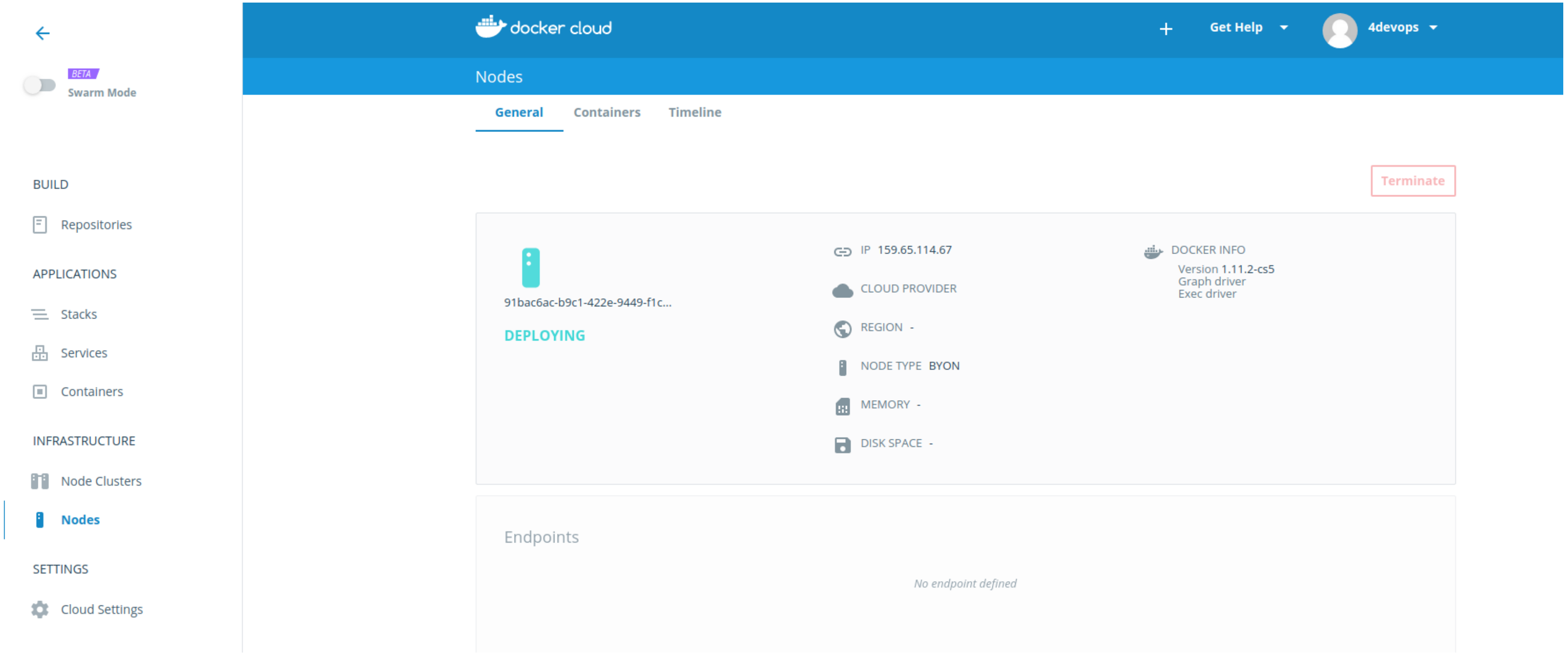
Let me spin up Digital Ocean nodes for You

- docker run -it 4devops/ssh
- ssh IP-ADDRESS
- For example: ssh 104.2.2.2
- curl -Ls https://get.cloud.docker.com/ | sudo -H sh -s 0b35c36f027148b98f34fde97769b523

DigitalOcean

- Linas- 159.65.126.44
- Tadas - 159.65.114.171
- Vaidotas - 159.65.114.45
- Aurelijus - 159.65.126.115
- Lolita – 159.65.124.113
- Mindaugas – 159.89.99.102
- Deimantas - 159.65.118.52
- Rimantas – 159.65.118.204
- Nerijus - 159.65.118.189
- Ignas - 159.65.126.175

Deploying Node



BETA
Swarm Mode

BUILD
Repositories
APPLICATIONS
Stacks
Services
Containers
INFRASTRUCTURE
Node Clusters
Nodes
SETTINGS
Cloud Settings

docker cloud
Get Help
4devops

Nodes

General | Containers | Timeline

Terminate

91bac6ac-b9c1-422e-9449-f1c...

DEPLOYING

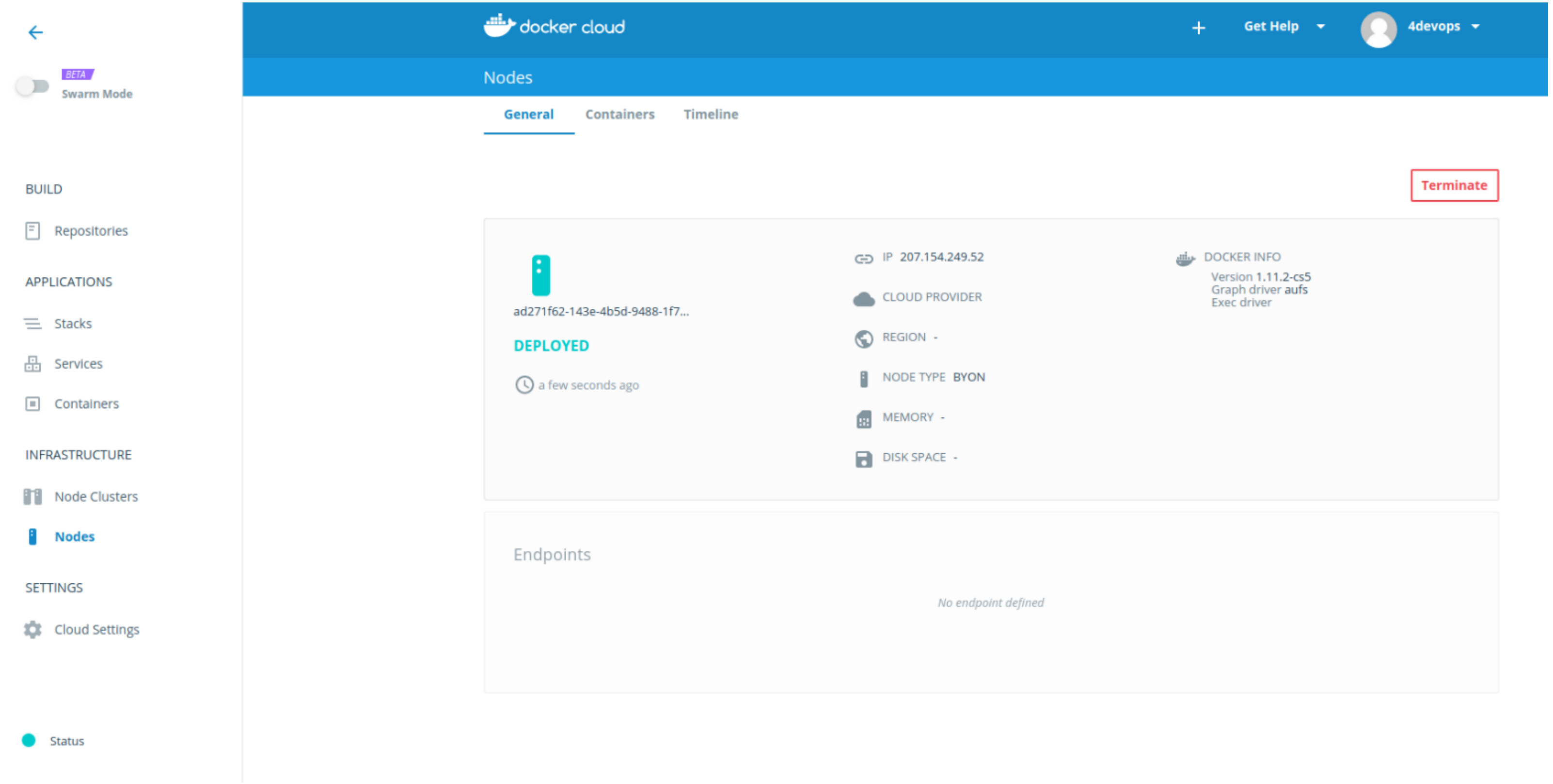
IP 159.65.114.67
CLOUD PROVIDER
REGION -
NODE TYPE BYON
MEMORY -
DISK SPACE -

DOCKER INFO
Version 1.11.2-cs5
Graph driver
Exec driver

Endpoints

No endpoint defined

Deployed Node



docker cloud

Get Help

4devops

BETA

Swarm Mode

BUILD

Repositories

APPLICATIONS

Stacks

Services

Containers

INFRASTRUCTURE

Node Clusters

Nodes

SETTINGS

Cloud Settings

Status

Nodes

General Containers Timeline

Terminate

ad271f62-143e-4b5d-9488-1f7...

DEPLOYED

a few seconds ago

IP 207.154.249.52

CLOUD PROVIDER

REGION -

NODE TYPE BYON

MEMORY -

DISK SPACE -

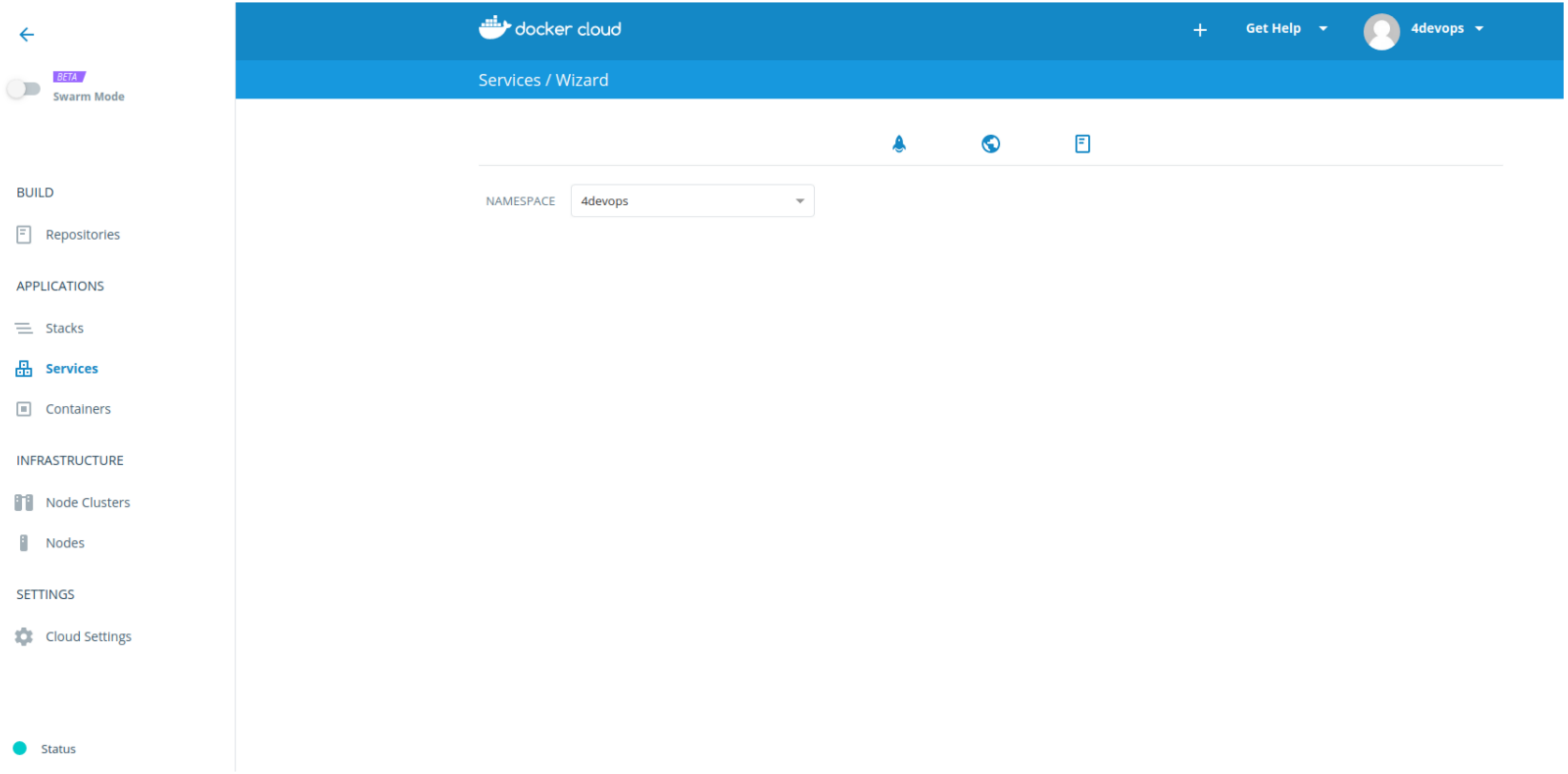
DOCKER INFO

Version 1.11.2-cs5
Graph driver aufs
Exec driver

Endpoints

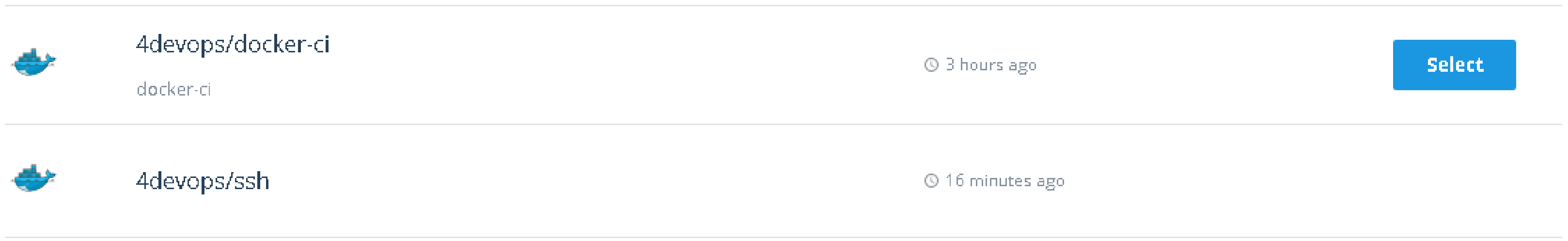
No endpoint defined

Create new service



NAMESPACE 4devops

Select repository


4devops/docker-ci

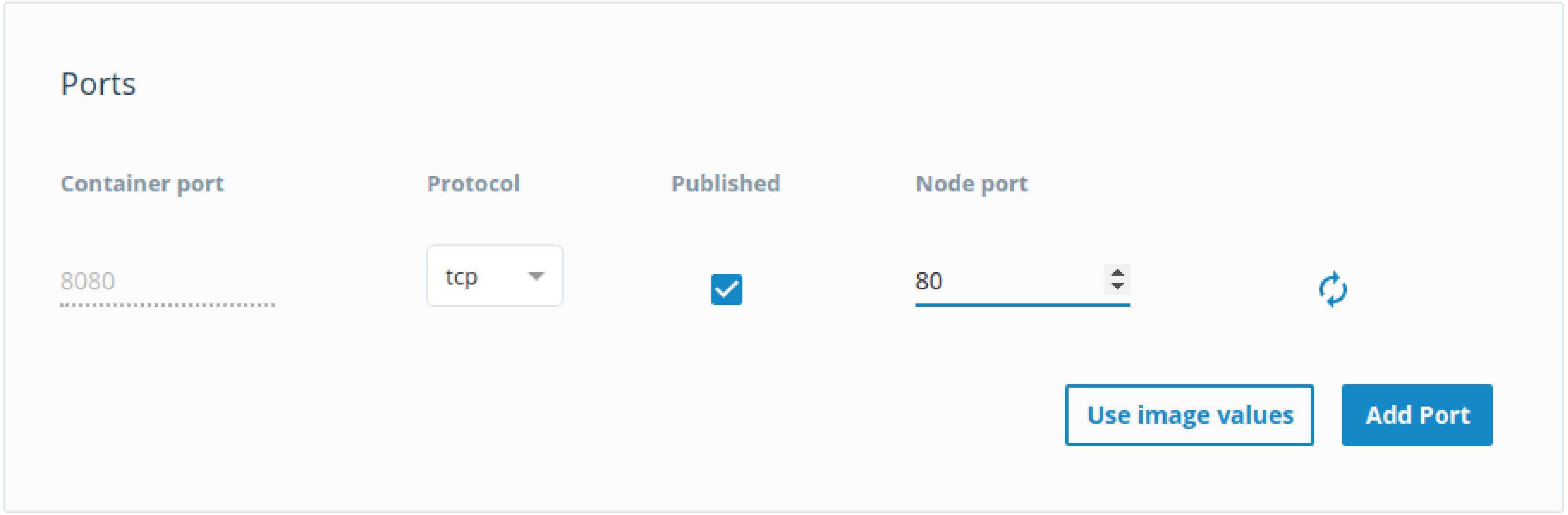
docker-ci

3 hours ago

4devops/ssh

16 minutes ago

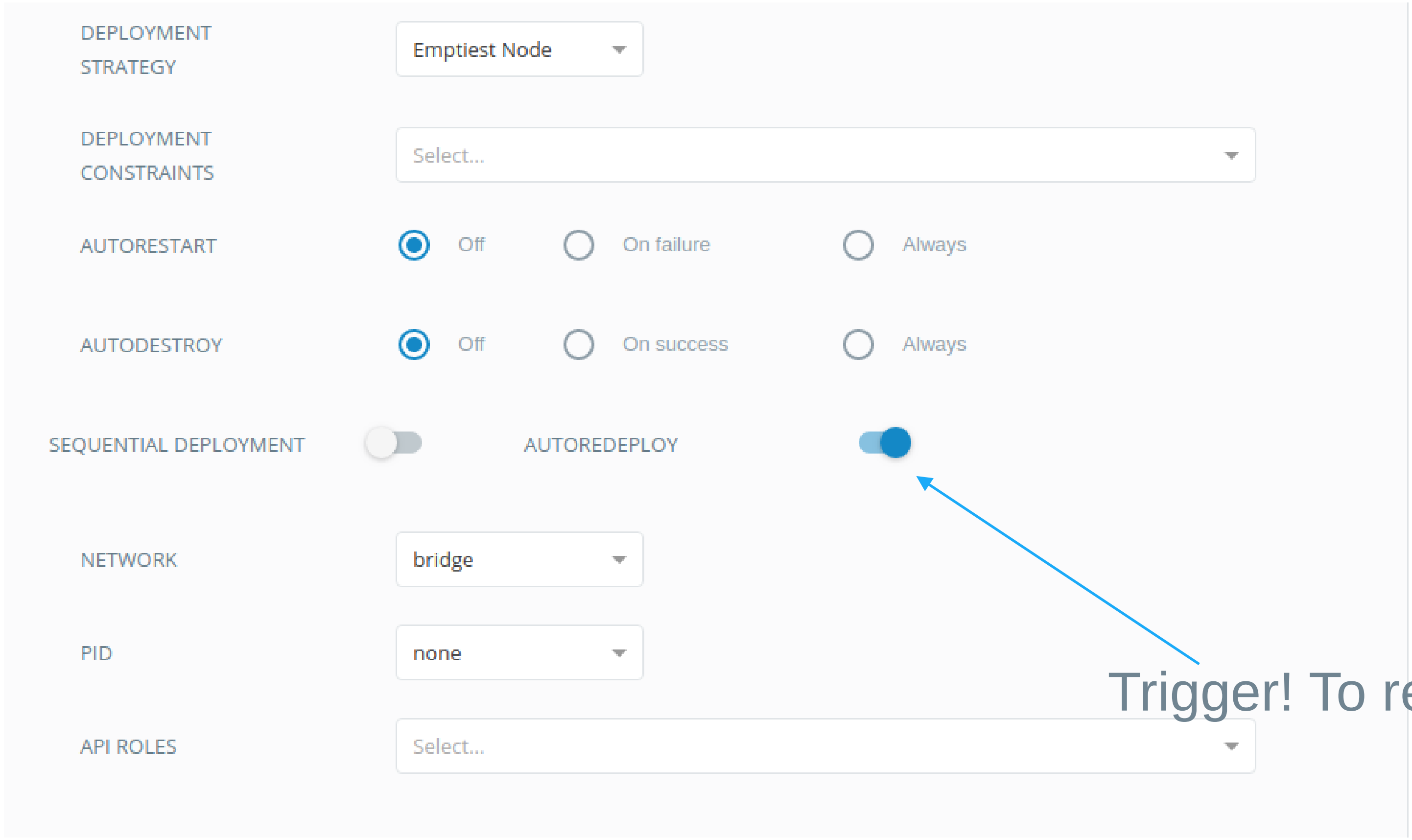
Expose port when creating service



Ports

Container port	Protocol	Published	Node port
8080	tcp	✓	80

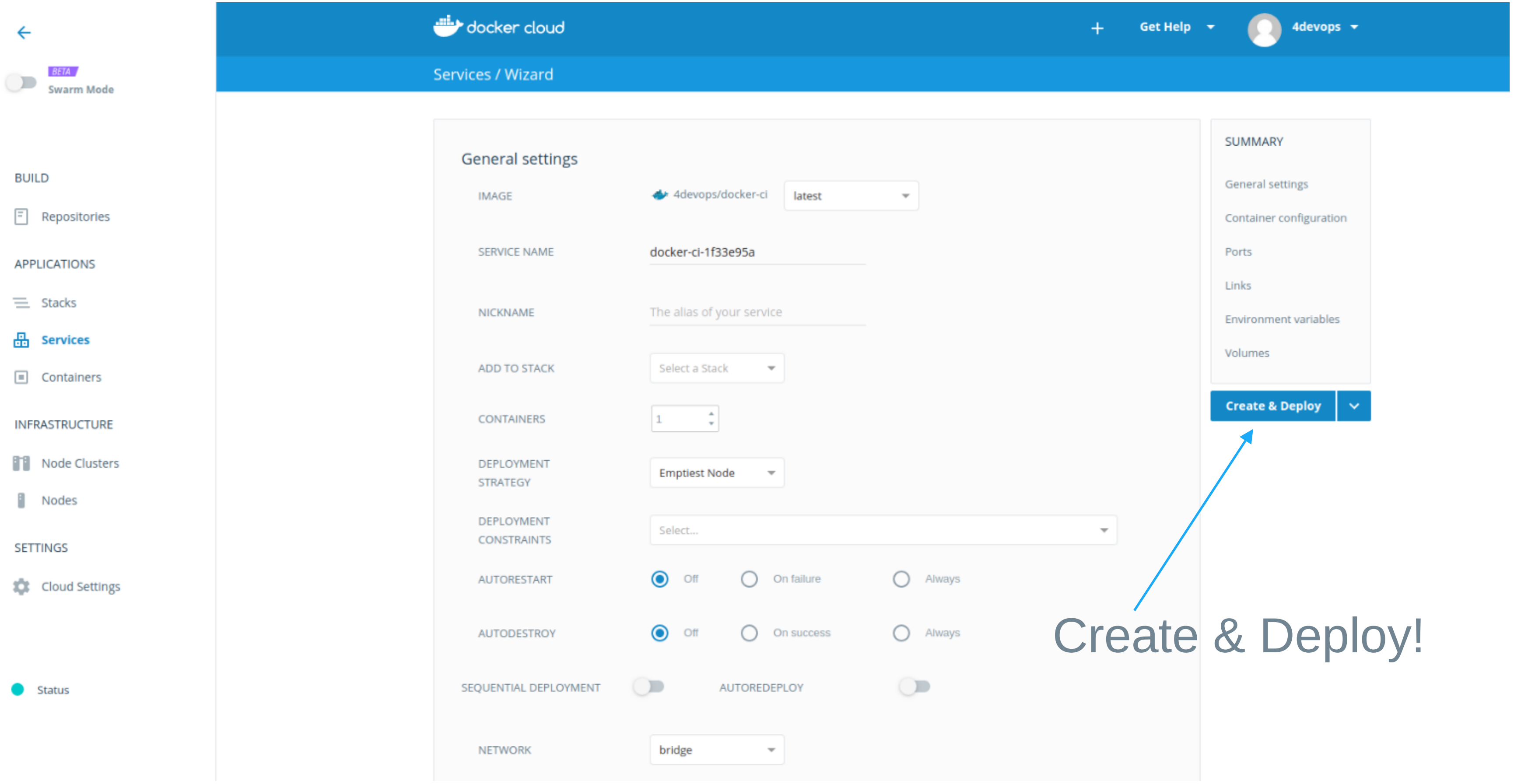
Create & Deploy



DEPLOYMENT STRATEGY	Emptiest Node
DEPLOYMENT CONSTRAINTS	Select...
AUTORESTART	[✓ Off] On failure Always
AUTODESTROY	[✓ Off] On success Always
SEQUENTIAL DEPLOYMENT	Off
AUTOREDEPLOY	On
NETWORK	bridge
PID	none
API ROLES	Select...

Trigger! To redeploy!

Create & Deploy



Services / Wizard

General settings

IMAGE: 4devops/docker-ci latest

SERVICE NAME: docker-ci-1f33e95a

NICKNAME: The alias of your service

ADD TO STACK: Select a Stack

CONTAINERS: 1

DEPLOYMENT STRATEGY: Emptiest Node

DEPLOYMENT CONSTRAINTS: Select...

AUTORESTART: [✓ Off] On failure Always

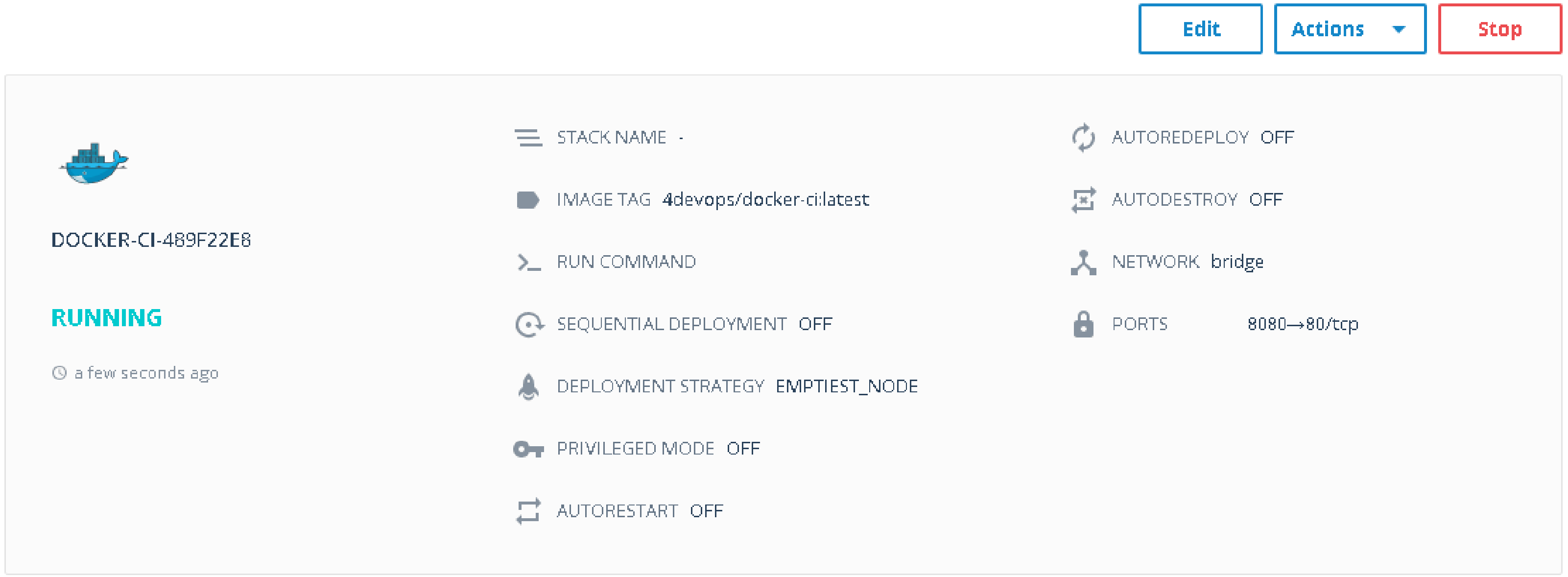
AUTODESTROY: [✓ Off] On success Always

SEQUENTIAL DEPLOYMENT AUTOREDEPLOY

NETWORK: bridge

Create & Deploy!

Start service



DOCKER-CI-489F22E8

RUNNING

a few seconds ago

STACK NAME -

IMAGE TAG 4devops/docker-ci:latest

RUN COMMAND

SEQUENTIAL DEPLOYMENT OFF

DEPLOYMENT STRATEGY EMPTIEST_NODE

PRIVILEGED MODE OFF

AUTORESTART OFF

AUTOREDEPLOY OFF

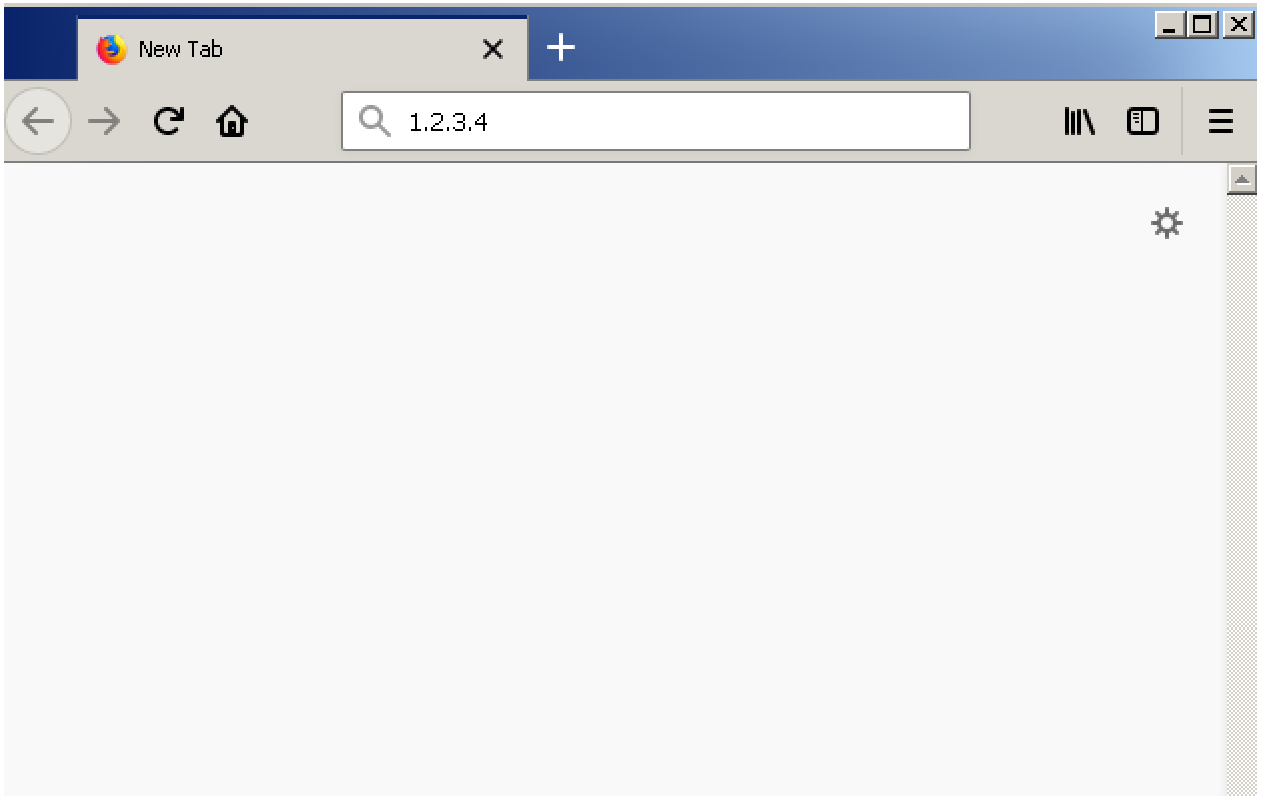
AUTODESTROY OFF

NETWORK bridge

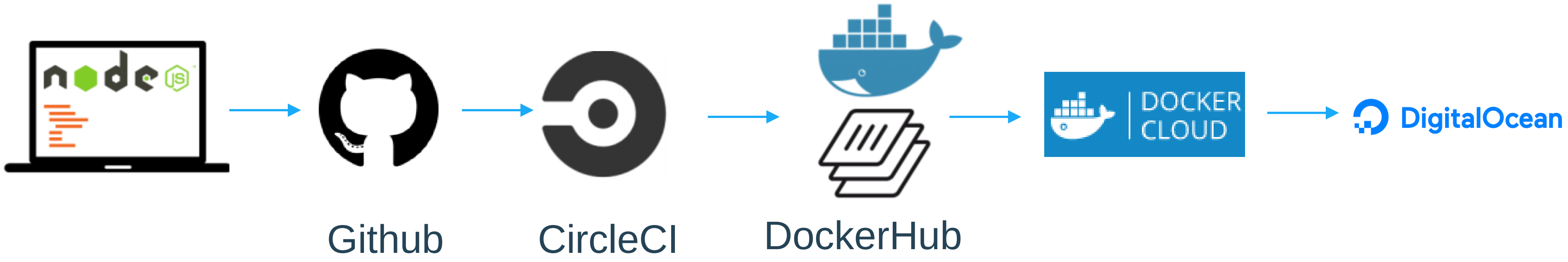
PORTS 8080→80/tcp

Check if You service is reachable

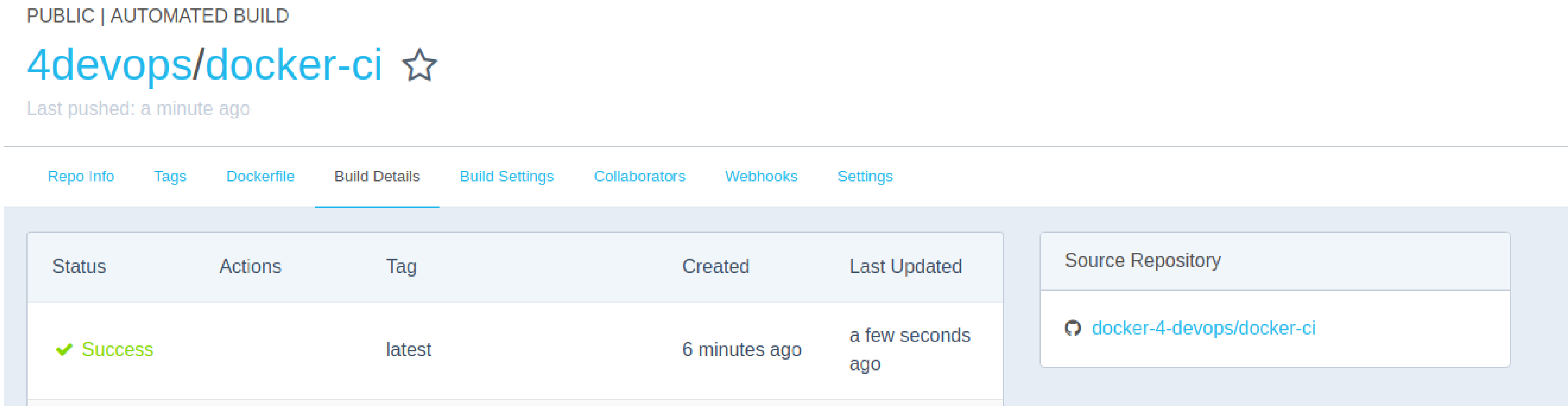
Use IP address Your node has



Let’s test our CI/CD workflow



Re-push code to DockerHub!



PUBLIC | AUTOMATED BUILD

4devops/docker-ci ☆

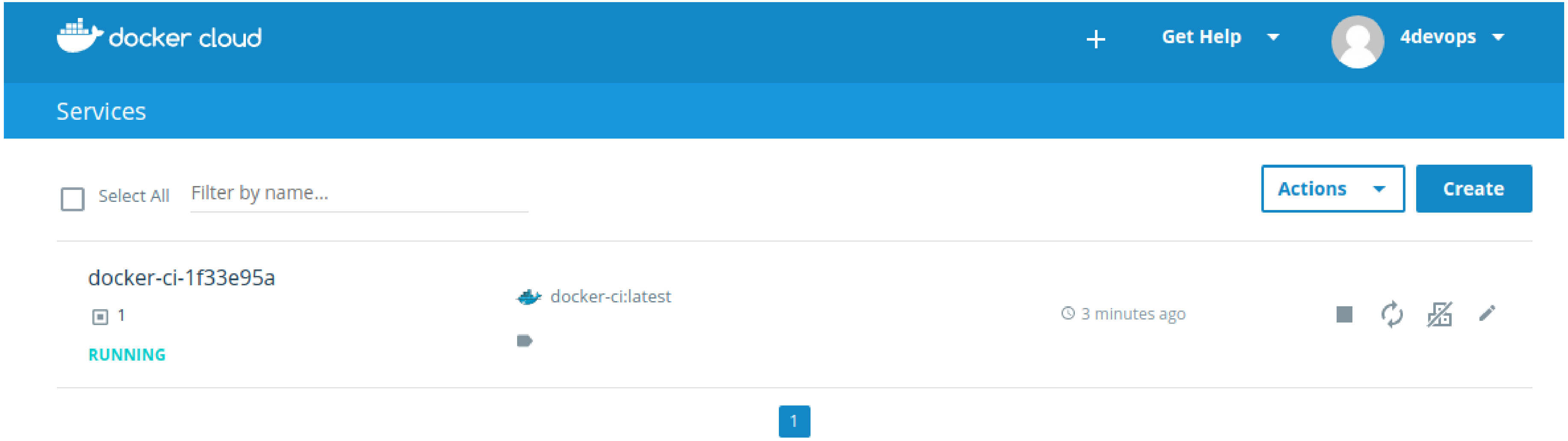
Last pushed: a minute ago

Status	Actions	Tag	Created	Last Updated
✓ Success		latest	6 minutes ago	a few seconds ago

Source Repository

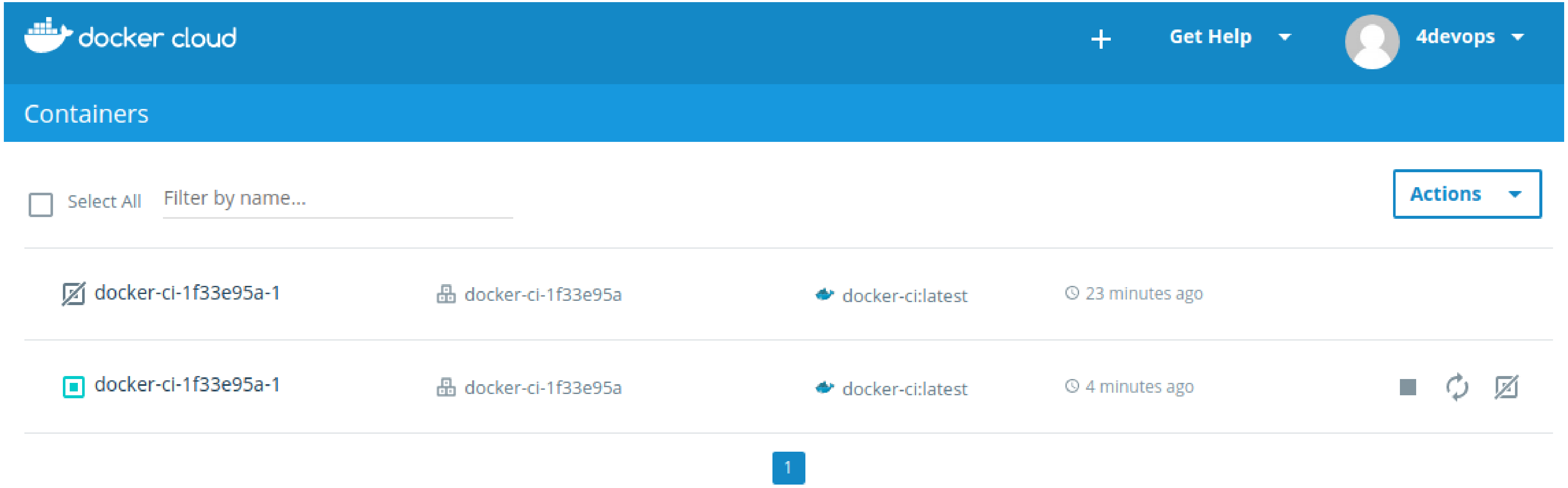
docker-4-devops/docker-ci

Docker Cloud will redeploy app automatically!



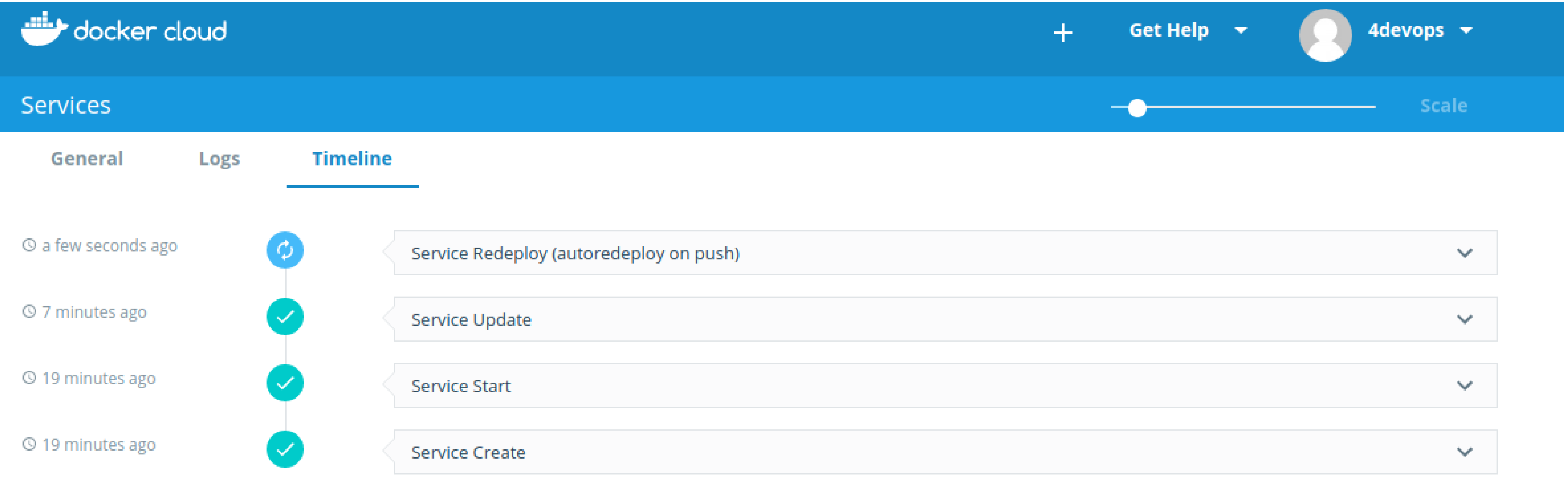
docker-ci-1f33e95a
1
RUNNING
docker-ci:latest
3 minutes ago

Docker Cloud will redeploy app automatically!



docker-ci-1f33e95a-1	docker-ci-1f33e95a	docker-ci:latest	23 minutes ago
docker-ci-1f33e95a-1	docker-ci-1f33e95a	docker-ci:latest	4 minutes ago

Timeline should show our continuous deployment



a few seconds ago — Service Redeploy (autoredeploy on push)

7 minutes ago — Service Update

19 minutes ago — Service Start

19 minutes ago — Service Create

docker