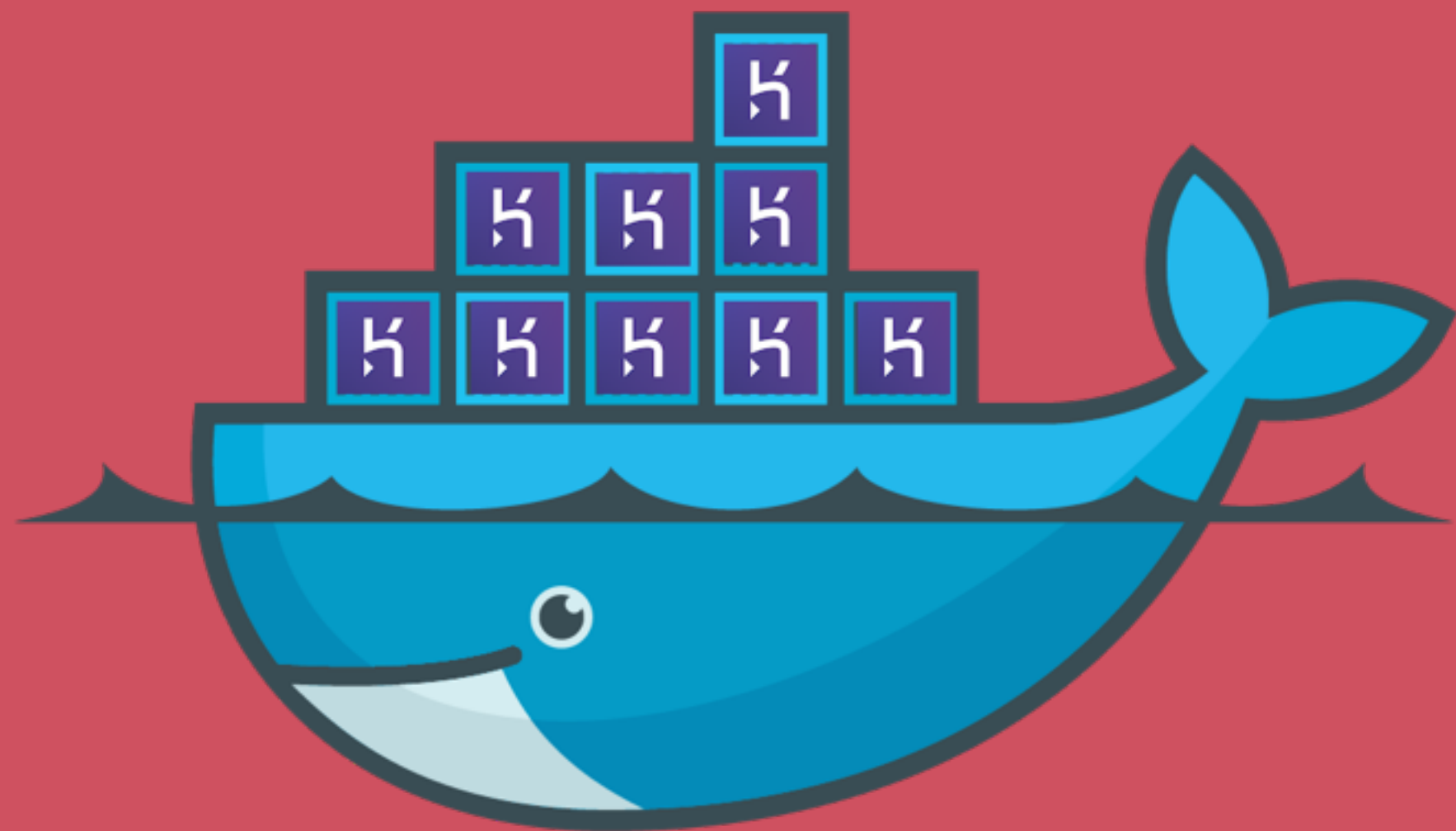


# Deploying Apps to Heroku with Docker



@brianfoshee

- Heroku
- Docker
- Heroku + Docker
- Demo

```
git push heroku master
```

```
$ git push heroku master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 293 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Ruby app detected
remote: -----> Compiling Ruby/Rails
remote: -----> Using Ruby version: ruby-2.2.2
remote: -----> Installing dependencies using 1.9.7
remote:          Running: bundle install --without development:test --path vendor/bundle --binstubs
remote: vendor/bundle/bin -j4 --deployment
remote: (...lots of gems installing...)
remote:          Bundle completed (0.78s)
remote:          Cleaning up the bundler cache.
remote: -----> Preparing app for Rails asset pipeline
remote:          Running: rake assets:precompile
remote:          Asset precompilation completed (1.62s)
remote:          Cleaning assets
remote:          Running: rake assets:clean
remote:
remote: ##### WARNING:
remote:          Include 'rails_12factor' gem to enable all platform features
remote:          See https://devcenter.heroku.com/articles/rails-integration-gems for more information.
remote:
remote: -----> Discovering process types
remote:          Procfile declares types => web
remote:          Default types for Ruby  => console, rake, worker
remote:
remote: -----> Compressing... done, 25.2MB
remote: -----> Launching... done, v13
remote:          https://pure-hamlet-9980.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy.... done.
To https://git.heroku.com/pure-hamlet-9980.git
   76a0e8d..eb6c92b  master -> master
```

```
$ git push heroku master
```

```
...
```

```
remote: Building source:
```

```
remote:
```

```
remote: -----> Ruby app detected
```

```
remote: -----> Compiling Ruby/Rails
```

```
remote: -----> Using Ruby version: ruby-2.2.2
```

```
...
```

GitHub

This repository

Search

Explore Features Enterprise Blog

Sign up

Sign in



heroku / heroku-buildpack-ruby

Watch 43

Star 416

Fork 1,211

Heroku's Ruby Buildpack for Cedar

1,136 commits

108 branches

123 releases

37 contributors



branch: master

heroku-buildpack-ruby / +



Merge pull request #391 from DyegoCosta/fix-detect



hone authored 17 days ago

latest commit 4bf2811456

bin	Allow path with spaces to be detected as Ruby	17 days ago
config	try using S3 instead of Cloudfront	2 years ago
lib	bump to v138	2 months ago
spec	Fix for bundler 1.9 regression	2 months ago
support/s3	s3 binaries	4 years ago
vendor	fixes very small typo	2 years ago
.gitignore	rev hatchet version	a year ago
.travis.yml	travis, again	9 months ago
CHANGELOG.md	[ci skip] v138 changelog	2 months ago
Gemfile	buildpack:tag task creates tag and pushes it	2 years ago
Gemfile.lock	[ci skip] fix buildpack release	3 months ago

Code

Issues 12

Pull requests 6

Pulse

Graphs

HTTPS clone URL

https://github.com/heroku-buildpack-ruby

You can clone with HTTPS or Subversion.

Clone in Desktop

Download ZIP



Buildpacks are the scripts that power app builds on Heroku. Check out the [Buildpacks article](#) for an overview of what role buildpacks play on the Heroku platform.

## Buildpack API



We encourage buildpack developers to use `sh` or `bash` to ensure compatibility with future Heroku stacks.

A buildpack consists of three scripts:

- `bin/detect`: Determines whether to apply this buildpack to an app.
- `bin/compile`: Used to perform the transformation steps on the app.
- `bin/release`: Provides metadata back to the runtime.

# Using a custom Buildpack



You can specify an exact version of a buildpack by using a git revision with the `buildpack:set` command.

```
git://repo.git#master  git://repo.git#v1.2.0
```

You can override the Heroku default buildpacks by specifying a custom buildpack with the `buildpacks:set` command:

```
$ heroku buildpacks:set https://github.com/heroku/heroku-buildpack-ruby
```





## Adding a buildpack

You can add additional buildpacks to your application with the `buildpacks:add` command. For example, if you need to add the Node.js buildpack to run a Grunt task you can add it with a command like this:

```
$ heroku buildpacks:add --index 1 https://github.com/heroku/heroku-buildpack-nodejs
```

This will insert the Node.js buildpack at the first position in the order of buildpack execution, and move the other buildpacks that are ahead of it down one position. Thus the Ruby buildpack will now be the second buildpack to run.

## Viewing buildpacks

You can view the complete list of buildpacks for an app by running this command:

```
$ heroku buildpacks
=== nameless-brushlands-4859 Buildpack
1. https://github.com/heroku/heroku-buildpack-nodejs.git
2. https://github.com/heroku/heroku-buildpack-ruby.git
```

```
$ git push heroku master
```

```
...
```

```
remote:      Bundle completed (0.78s)
```

```
remote:      Cleaning up the bundler cache.
```

```
remote: -----> Preparing app for Rails asset pipeline
```

```
remote:      Running: rake assets:precompile
```

```
remote:      Asset precompilation completed (1.62s)
```

```
remote:      Cleaning assets
```

```
remote:      Running: rake assets:clean
```

```
remote:
```

```
remote: ##### WARNING:
```

```
remote:      Include 'rails_12factor' gem to enable all platform features
```

```
remote:      See https://devcenter.heroku.com/articles/rails-integration-
```

```
gems for more information.
```

```
remote:
```

```
remote: -----> Discovering process types
```

```
remote:      Procfile declares types -> web
```

```
remote:      Default types for Ruby -> console, rake, worker
```

```
remote:
```

```
remote: -----> Compressing... done, 25.2MB
```

```
remote: -----> Launching... done, v13
```

```
remote:      https://pure-hamlet-9980.herokuapp.com/ deployed to Heroku
```

```
...
```

```
docker run -it heroku/cedar:14 /bin/bash
```



hub.docker.com/r/heroku/cedar/~/.dockerfile/



# heroku/cedar ☆

Last pushed: 15 days ago

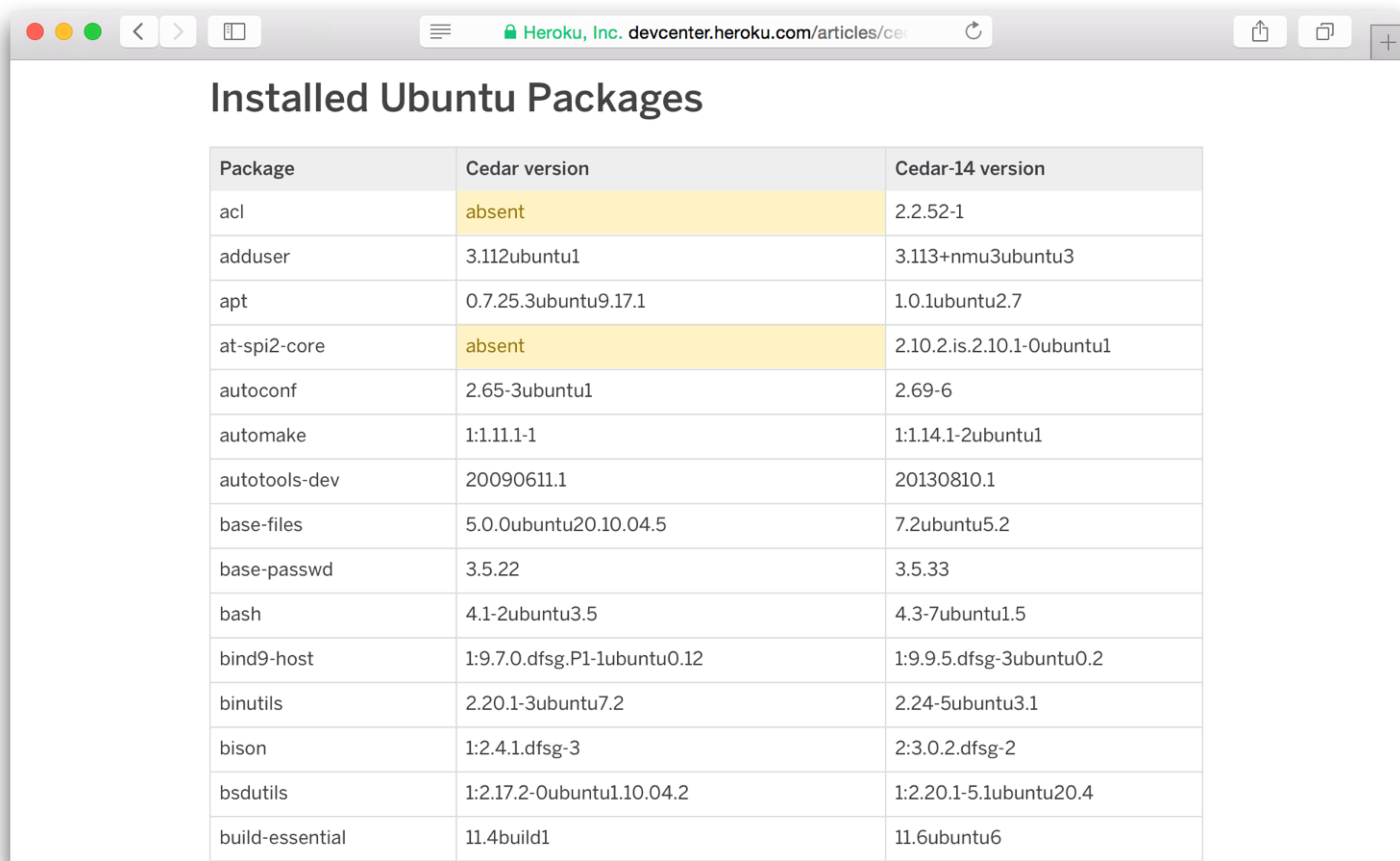
[Repo Info](#)

[Tags](#)

[Dockerfile](#)

[Build Details](#)

```
FROM ubuntu-debootstrap:14.04
COPY ./bin/cedar-14.sh /tmp/build.sh
RUN LC_ALL=C DEBIAN_FRONTEND=noninteractive /tmp/build.sh \
    && rm -rf /var/lib/apt/lists/*
```



Package	Cedar version	Cedar-14 version
acl	absent	2.2.52-1
adduser	3.112ubuntu1	3.113+nmu3ubuntu3
apt	0.7.25.3ubuntu9.17.1	1.0.1ubuntu2.7
at-spi2-core	absent	2.10.2.is.2.10.1-0ubuntu1
autoconf	2.65-3ubuntu1	2.69-6
automake	1:1.11.1-1	1:1.14.1-2ubuntu1
autotools-dev	20090611.1	20130810.1
base-files	5.0.0ubuntu20.10.04.5	7.2ubuntu5.2
base-passwd	3.5.22	3.5.33
bash	4.1-2ubuntu3.5	4.3-7ubuntu1.5
bind9-host	1:9.7.0.dfsg.P1-1ubuntu0.12	1:9.9.5.dfsg-3ubuntu0.2
binutils	2.20.1-3ubuntu7.2	2.24-5ubuntu3.1
bison	1:2.4.1.dfsg-3	2:3.0.2.dfsg-2
bsdutils	1:2.17.2-0ubuntu1.10.04.2	1:2.20.1-5.1ubuntu20.4
build-essential	11.4build1	11.6ubuntu6

~1000 packages in total



Automated generation of stack images

83 commits

3 branches

3 releases

8 contributors

branch: master

stack-images / +

Merge branch 'install-stunnel-package'		
<div>tt</div>	authored on Jun 10	latest commit f26925ad47
<div>bin</div>	Install stunnel package	a month ago
<div>debian</div>	Initial commit.	2 years ago
<div>.gitignore</div>	Initial commit.	2 years ago
<div>Dockerfile</div>	resolves syncing with build script and brings into parity with cedari...	4 months ago
<div>INSTALL.md</div>	new readme and move build instructions to separate file	3 months ago
<div>README.md</div>	new readme and move build instructions to separate file	3 months ago
<div>Vagrantfile</div>	Upgrade Vagrant box	a year ago

<> Code

Issues

3

Pull requests

3

Pulse

Graphs

HTTPS clone URL

https://github.com/hero

You can clone with [HTTPS](#) or [Subversion](#). ?

Clone in Desktop

Download ZIP





hub.docker.com/r/heroku/ruby/~/.dockerfile/



# heroku/ruby

Last pushed: an hour ago

[Repo Info](#)

[Tags](#)

[Dockerfile](#)

[Build Details](#)

```
FROM heroku/cedar:14
MAINTAINER Terence Lee <terence@heroku.com>

RUN mkdir -p /app/user
WORKDIR /app/user

ENV GEM_PATH /app/heroku/ruby/bundle/ruby/2.2.0
ENV GEM_HOME /app/heroku/ruby/bundle/ruby/2.2.0
RUN mkdir -p /app/heroku/ruby/bundle/ruby/2.2.0

# Install Ruby
RUN mkdir -p /app/heroku/ruby/ruby-2.2.3
RUN curl -s --retry 3 -L https://heroku-buildpack-ruby.s3.amazonaws.com/cedar-14/ruby-2
ENV PATH /app/heroku/ruby/ruby-2.2.3/bin:$PATH
```

Docker does not run on Heroku

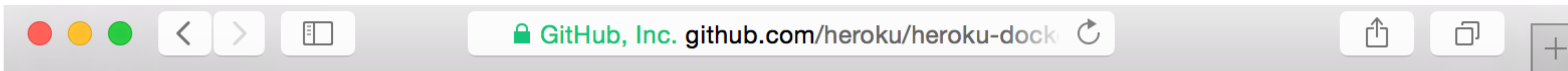
# Creating builds



Need a place to upload source tarballs for use with the Build API? See the [sources endpoint](#) section below.

Creating a build from a source tarball is simple:

```
$ curl -n -X POST https://api.heroku.com/apps/example-app/builds \
-d '{"source_blob":{"url":"https://github.com/heroku/node-js-sample/archive
-H 'Accept: application/vnd.heroku+json; version=3' \
-H "Content-Type: application/json"
{
  "created_at": "2014-04-23T02:47:04+00:00",
  "id": "01234567-89ab-cdef-0123-456789abcdef",
  "source_blob": {
    "url": "https://github.com/heroku/node-js-sample/archive/cb6999d361a024
    "version": "cb6999d361a0244753cf89813207ad53ad906a14"
  },
  "slug": {
    "id": null
  },
  "status": "pending",
  "updated_at": "2014-04-23T02:47:11+00:00",
  "user": {
    "email": "username@example.com",
    "id": "01234567-89ab-cdef-0123-456789abcdef"
  }
}
```



# Heroku Docker CLI plugin

---

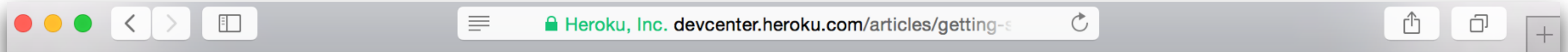
Heroku Toolbelt plugin to help configure, test and release apps to Heroku using Docker and Docker Compose.

## Installation

---

```
$ heroku plugins:install heroku-docker
```





# Release to Heroku

At any time, you can create a slug from your local container and release to a Heroku app (like the one create with `heroku create` above):

```
$ heroku docker:release
...
$ heroku open
```

Because we used the Heroku app's database for local development in the previous step, the schema should be ready.

```
function createLocalSlug() {
  cli.log('creating local slug...');
  try {
    var slugPath = os.tmpdir();
    var imageId = docker.ensureStartImage(context.cwd);
    if (!imageId) return Promise.reject(new Error('Unable to find a start image'));

    var containerId = child.execSync(`docker run -d ${imageId} tar cfvz /tmp/slug.tgz -C / --exclude=.git ./app`,
      encoding: 'utf8'
    ).trim();
    child.execSync(`docker wait ${containerId}`);
    child.execSync(`docker cp ${containerId}:/tmp/slug.tgz ${slugPath}`);
    child.execSync(`docker rm -f ${containerId}`);
    return Promise.resolve(path.join(slugPath, 'slug.tgz'));
  }
  catch (e) {
    return Promise.reject(e);
  }
}
```

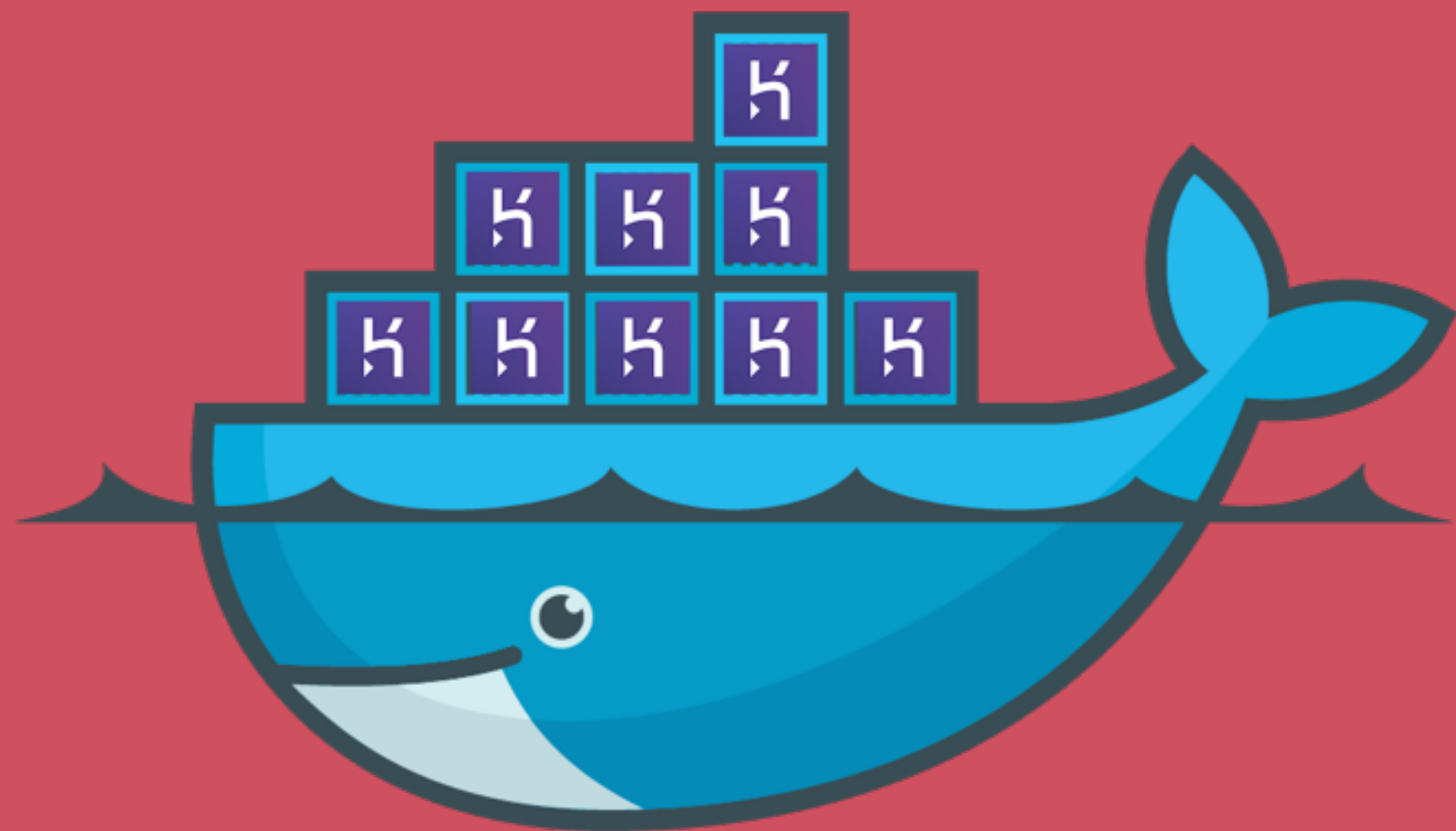


Demo

- Control over app environment
- Dev/Prod parity
- Probably not particularly useful locally
- More useful when integrated with CI

- <https://devcenter.heroku.com/categories/language-support>
- <https://devcenter.heroku.com/articles/cedar-ubuntu-packages>
- <https://devcenter.heroku.com/articles/buildpacks>
- <https://devcenter.heroku.com/articles/buildpack-api>
- <https://devcenter.heroku.com/articles/using-multiple-buildpacks-for-an-app>
- <https://github.com/heroku/stack-images>
- <https://registry.hub.docker.com/u/heroku/cedar>
- <https://devcenter.heroku.com/articles/docker>
- <https://github.com/heroku/heroku-docker>
- <https://devcenter.heroku.com/articles/build-and-release-using-the-api>
- [https://blog.heroku.com/archives/2013/12/20/  
programmatically\\_release\\_code\\_to\\_heroku\\_using\\_the\\_platform\\_api](https://blog.heroku.com/archives/2013/12/20/programmatically_release_code_to_heroku_using_the_platform_api)

# Questions?



@brianfoshee