

 無料電子ブック

学習

heroku

Free unaffiliated eBook created from
Stack Overflow contributors.

#heroku

.....	1
1:	2
.....	2
Examples.....	2
.....	2
.....	2
.....	2
Debian / Ubuntu	2
Heroku Toolbelt.....	2
.....	2
.....	3
.....	3
Heroku	3
.....	3
.....	3
Heroku.....	3
2: Heroku node.js Hello World	5
.....	5
Examples.....	5
Heroku node.js hello world.....	5
3: Heroku Postgres	7
Examples.....	7
HerokuPostgres.....	7
heroku.....	7
4: Heroku	8
.....	8
Examples.....	8
Heroku.....	8
5: Heroku	9
.....	9
.....

9	
Examples	10
H10 -	10
H11 -	10
H12 -	10
H13 -	11
H14 -	11
H15 -	11
6:	12
.....	12
.....	12
Examples	12
.....	12
OS X	12
Windows	12
Debian / Ubuntu	12
.....	12
.....	13
.....	13
7:	14
.....	14
.....	14
Examples	14
CLI	14
8:	16
Examples	16
.....	16
.....	16
9:	18
Examples	18
Heroku	18

10:	19
.....	19
Examples	19
.....	19
.....	19
.....	20
.....	20
.....	20
11:	22
.....	22
Examples	22
.....	22
12:	23
.....	23
Examples	23
Git	23
git	23
Heroku	23
.....	23
.....	25

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [heroku](#)

It is an unofficial and free heroku ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official heroku.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: ヒロクをめよう

Herokuは、チームがチームなしでWebアプリケーションをにできるようにする、あるPaaS Platform-as-a-Serviceプロバイダーです。 Herokuは2007、[Salesforce](#)によってされています。

このセクションでは、Herokuのと、がそれをいたいについてします。

また、Herokuのきなについてもし、するトピックにリンクするがあります。 Herokuのドキュメントはしいものなので、それらのトピックのバージョンをするがあります。

Examples

インストールまたはセットアップ

Herokuアプリをローカルでしてするには、Heroku Toolbeltがです。ここにはいくつかのがあります。

ダウンロード

HerokuのWebサイトから[Heroku Toolbelt](#)インストーラーをダウンロードしてください。

herokuをbrewインストールbrew

```
brew install heroku
```

Debian / Ubuntu

このスクリプトをします。

```
wget -O- https://toolbelt.heroku.com/install-ubuntu.sh | sh
```

このスクリプトは、Herokuリポジトリをaptにし、Herokuリリースキーをインストールし、Heroku Toolbeltをインストールし、にじてRubyをインストールします。

のスク립トとじようにオンラインとパイプをbashにつけると、まずソースをむことをくおめします。

Heroku Toolbeltをする

アプリケーションをする

```
heroku create your-app-name
```

へロクに

```
git push heroku master
```

ブラウザでアプリケーションをく

```
heroku open your-app-name
```

Herokuのコマンドをする

```
heroku commands
```

なヘルプ

```
heroku help
```

のコマンドのヘルプ

```
heroku help <command>
```

Herokuアプリケーションの

`heroku create` コマンドをしてHerokuアプリケーションをすることができます。 Herokuにデプロイするアプリケーションには、のコードベース、、アドオンなどがあります。

Herokuアプリケーションは、グローバルにのをちます。にがけられているHerokuアプリケーションをしようとする、エラーがします。

しいHerokuアプリケーションをするはのとおりです。

```
heroku create [app_name]
```

`heroku create`するときアプリケーションをしないと、Herokuはランダムなアプリケーションをします。

HerokuアプリケーションをするがあるAmazonリージョンをすることもできます。デフォルトでは、すべてのHerokuアプリケーションは`us`にされます。リージョンをしたいは、のようにアプリケーションをしてリージョンをできます。

```
heroku create [app_name] --region eu
```

、は2つしかありません `us`と`eu` ヨーロッパ。

オンラインでヒロクをめようをむ <https://riptutorial.com/ja/heroku/topic/959> ヒロクをめよう

2: Heroku node.js Hello World

ログイン

```
heroku login
```

アプリを作る

```
heroku create または heroku create your_name
```

をする

```
git clone https://github.com/zoutepopcorn/herokuworld
cd herokuworld
```

あなたのブラウザで**app**にアクセスする

```
https://your_name.herokuapp.com/
```

オプションでローカルにテストする

```
heroku local web
```

チェックlolhost5000

では、のnode.jsアプリとはがうのですか package.json

```
"scripts": {
  "start": "node index.js"
},
"engines": {
  "node": "7.6.0"
}
```

index.js

```
process.env.PORT
```

ローカルポート5000。HerokuはあなたのアプリのURLのポート80にマップします。

Examples

Heroku node.js hello world

index.js

```
var http = require("http");
```

```
http.createServer(function(request, response) {
  response.writeHead(200, {"Content-Type": "text/plain"});
  response.write("Heroku world!");
  response.end();
}).listen(process.env.PORT);
```

package.json

```
{
  "name": "node-example",
  "version": "1.0.0",
  "description": "Hello world Heroku",
  "scripts": {
    "start": "node index.js"
  },

  "keywords": [
    "example",
    "heroku"
  ],
  "author": "Johan",
  "license": "MIT",
  "engines": {
    "node": "7.6.0"
  }
}
```

オンラインでHeroku node.js Hello Worldをむ <https://riptutorial.com/ja/heroku/topic/9897/heroku-node-js-hello-world>

3: Heroku Postgres

Examples

HerokuでPostgresデータベースをリセットする

Herokuでデータベースをリセットする

1. **SHARED_DATABASE_URL**をする、データベースをします。

```
heroku pg:reset DATABASE
```

2. データベースをもせずにします。

```
heroku run rake db:migrate
```

3. データベースにシードデータをします。

```
heroku run rake db:seed
```

2と3を1つのコマンドにまとめるには、のようになります。

```
heroku run rake db:setup
```

herokuデータベースをローカルデータベースにコピーする

herokuデータベースをローカルデータベースにコピーする

1. でコピープロセスをします。

```
heroku pg:pull DATABASE_URL change_to_your_data_base_name --app change_to_your_app_name
```

2. のクエリをしてdbのをします。

```
GRANT ALL PRIVILEGES ON DATABASE change_to_your_data_base_name to change_to_your_user; ALTER DATABASE change_to_your_data_base_name OWNER TO change_to_your_user;
```

3. データベースのすべてのテーブルのクエリをしてします。

```
SELECT 'ALTER TABLE ' || schemaname || '.' || tablename || ' OWNER TO change_to_your_user;' FROM pg_tables WHERE NOT schemaname IN ('pg_catalog', 'information_schema') ORDER BY schemaname, tablename;
```

オンラインでHeroku Postgresをむ <https://riptutorial.com/ja/heroku/topic/6239/heroku-postgres>

4: Heroku アドオン

き

Herokuでなさまざまなアドオンにするとその

Examples

Heroku スケジューラ

Heroku Schedulerのインストール

```
heroku addons:create scheduler:standard
```

オンラインでHerokuアドオンをむ <https://riptutorial.com/ja/heroku/topic/8906/herokuアドオン>

5: Heroku エラーコード

き

アプリケーションにエラーがすると、HerokuはHTTPステータスコード503のエラーページをします。ただし、エラーをデバッグするのにつよう、プラットフォームにはカスタムエラーもログにされます。タイプのエラーはのエラーコードをし、すべてのHTTPエラーはHでまり、すべてのランタイムエラーはRでまります。ロギングエラーはLでまります。

- H10 - アプリクラッシュ
- H11 - バックログがすぎます
- H12 - のタイムアウト
- H13 - なしでがじられました
- H14 - ウェブダイノスはされていません
- H15 - アイドル
- H16 - herokuapp.comにリダイレクト
- H17 - フォーマットされていないHTTPレスポンス
- H18 - サーバーがされました
- H19 - バックエンドのタイムアウト
- H20 - Appブートタイムアウト
- H21 - バックエンドがされました
- H22 - にしました
- H23 - エンドポイントがってされている
- H24 -
- H25 - HTTP
- H26 - リクエストエラー
- H27 - クライアントがされました
- H28 - クライアントアイドル
- H80 - メンテナンスモード
- H81 - ブランクアプリ
- H82 - なdynoりてがいたされた
- H99 - プラットフォームエラー
- R10 - ブートタイムアウト
- R12 - タイムアウト
- R13 - アタッチエラー
- R14 - メモリクォータをえました
- R15 - メモリのクォータをにえました
- R16 - デタッチ
- R17 - チェックサムエラー
- R99 - プラットフォームエラー
- L10 - ドレインバッファオーバーフロー

- L11 - テールバッファオーバーフロー
- L12 - ローカルバッファオーバーフロー
- L13 - ローカルエラー
- L14 - のエラー

Examples

H10 - アプリクラッシュ

クラッシュしたWeb dynoまたはWeb dynoのブートタイムアウトがこのエラーをします。

```
2010-10-06T21:51:04-07:00 heroku[web.1]: State changed from down to starting
2010-10-06T21:51:07-07:00 app[web.1]: Starting process with command: `bundle exec rails server -p 22020`
2010-10-06T21:51:09-07:00 app[web.1]: >> Using rails adapter
2010-10-06T21:51:09-07:00 app[web.1]: Missing the Rails 2.3.5 gem. Please `gem install -v=2.3.5 rails`, update your RAILS_GEM_VERSION setting in config/environment.rb for the Rails version you do have installed, or comment out RAILS_GEM_VERSION to use the latest version installed.
2010-10-06T21:51:10-07:00 heroku[web.1]: Process exited
2010-10-06T21:51:12-07:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/" host=myapp.herokuapp.com fwd=17.17.17.17 dyno= connect= service= status=503 bytes=
```

H11 - バックログがすぎます

アプリケーションができるよりもくHTTPリクエストがすると、のルータにきなバックログをすることができます。のルータのバックログがしきいをえると、ルータはアプリケーションがボリュームにいついていないとします。バックログがこのサイズをえているり、ごとにH11エラーがされます。このしきいのなは、アプリのダイノス、々のリクエストの、およびアプリののリクエストなど、さまざまによってします。

```
2010-10-06T21:51:07-07:00 heroku[router]: at=error code=H11 desc="Backlog too deep" method=GET path="/" host=myapp.herokuapp.com fwd=17.17.17.17 dyno= connect= service= status=503 bytes=
```

は、dynosをしたり、データベースをしたりたとえば、インデックスをするなど、コードをすることで、アプリのスループットをさせることです。いつものように、パフォーマンスのはアプリケーションのものであり、プロファイリングがです。

H12 - のタイムアウト

HTTPがするまでに30かかりました。のでは、Railsアプリケーションはページのレンダリングに37かかります。HTTPルータはサイクルをするに503をしますが、Railsプロセスはし、ルータメッセージのにメッセージがされます。

```
2010-10-06T21:51:07-07:00 app[web.2]: Processing PostController#list (for 75.36.147.245 at 2010-10-06 21:51:07) [GET]
2010-10-06T21:51:08-07:00 app[web.2]: Rendering template within layouts/application
```

```
2010-10-06T21:51:19-07:00 app[web.2]: Rendering post/list
2010-10-06T21:51:37-07:00 heroku[router]: at=error code=H12 desc="Request timeout" method=GET
path="/" host=myapp.herokuapp.com fwd=17.17.17.17 dyno=web.1 connect=6ms service=30001ms
status=503 bytes=0
2010-10-06T21:51:42-07:00 app[web.2]: Completed in 37000ms (View: 27, DB: 21) | 200 OK
[http://myapp.herokuapp.com/]
```

この30のは、ルータによってされ、カーネルのキューとアプリケーションをめて、`dyno`でやされたすべてのをみます。

H13 - なしでがじられました

このエラーは、Web `dyno`のプロセスがをけるが、もきまずにソケットをじるとスローされます。

```
2010-10-06T21:51:37-07:00 heroku[router]: at=error code=H13 desc="Connection closed without
response" method=GET path="/" host=myapp.herokuapp.com fwd=17.17.17.17 dyno=web.1
connect=3030ms service=9767ms status=503 bytes=0
```

これがこるのあるの1つは、Unicorn Webサーバーが30のタイムアウトでされ、タイムアウトがするにワーカによってがされなかったです。この、Unicornはデータがきまれるにをじ、としてH13となります。

H14 - ウェブダイノスはされていません

これはウェブダイノスを0`dynos`にスケールダウンしたがいです。それをするには、Webダイノスを1つの`dynos`にスケールアップします。

```
$ heroku ps:scale web=1
```

`heroku ps`コマンドをして、Webダイノスのをします。

```
2010-10-06T21:51:37-07:00 heroku[router]: at=error code=H14 desc="No web processes running"
method=GET path="/" host=myapp.herokuapp.com fwd=17.17.17.17 dyno= connect= service=
status=503 bytes=
```

H15 - アイドル

`dyno`はなをせず、55のアクティブのためにしました。えは、レスポンスはにされなかった50バイトの`Content-Length`をしました。

```
2010-10-06T21:51:37-07:00 heroku[router]: at=error code=H15 desc="Idle connection" method=GET
path="/" host=myapp.herokuapp.com fwd=17.17.17.17 dyno=web.1 connect=1ms service=55449ms
status=503 bytes=18
```

オンラインでHerokuエラーコードをむ <https://riptutorial.com/ja/heroku/topic/8321/herokuエラーコード>

6: コマンドライン

き

はHeroku ToolbeltとしてられていたHerokuコマンドラインインターフェイスCLIは、さまざまなオペレーティングシステムのコマンドライン/シェルからHerokuアプリケーションをおよびするためのツールです。

- `$ heroku --version`
- `$ heroku login`
- `$ heroku create`

Examples

ダウンロードとインストール

OS X

OS Xインストーラをダウンロードしてします。

Windows

Windowsインストーラを32ビット 64ビットでダウンロードしてします。

Debian / Ubuntu

aptリポジトリをしてCLIをインストールするには、のコマンドをします。

```
$ sudo add-apt-repository "deb https://cli-assets.heroku.com/branches/stable/apt ./"
$ curl -L https://cli-assets.heroku.com/apt/release.key | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install heroku
```

スタンドアロン

あなたのPATHからバイナリにアクセスできるように、tarballをダウンロードしてします。えは

```
$ echo replace OS/ARCH with values as noted below
$ wget https://cli-assets.heroku.com/branches/stable/heroku-OS-ARCH.tar.gz
```

```
$ tar -xvzf heroku-OS-ARCH /usr/local/lib/heroku
$ ln -s /usr/local/lib/heroku/bin/heroku /usr/local/bin/heroku
```

インストールをする

CLIのインストールをするには、`heroku --version` コマンドをします。

```
$ heroku --version
heroku-cli/5.6.0-010a227 (darwin-amd64) go1.7.4
```

にコマンドをすると、Herokuのをするようめられます。あなたのメールアドレスとAPIトークンは`~/.netrc`され、ののために`~/.netrc`されます。

には、Heroku CLIをインストールしたにをしてログインするとよいでしょう。gitを使ってHerokuアプリケーションリポジトリをプッシュまたはクローンすることができます

```
$ heroku login
Enter your Heroku credentials.
Email: adam@example.com
Password (typing will be hidden):
Authentication successful.
```

これで、のHerokuアプリをするがいました。

```
$ cd ~/myapp
$ heroku create
Creating app... done, ☐ sleepy-meadow-81798
https://sleepy-meadow-81798.herokuapp.com/ | https://git.heroku.com/sleepy-meadow-81798.git
```

オンラインでコマンドラインをむ <https://riptutorial.com/ja/heroku/topic/8324/コマンドライン>

7: パイプライン

- herokuパイプライン<install | create | promote> ...

パイプラインは、じコードベースをするHerokuアプリケーションのグループです。パイプラインのアプリケーションは、なワークフローでなるをす「レビュー」、「」、「ステージング」、および「プロダクション」にされます。

Examples

CLIによるパイプライン

パイプラインのインストール

Heroku Toolbeltがインストールされたら、 [Pipelinesプラグイン](#) もです。

```
heroku plugins:install heroku-pipelines
```

パイプラインの

パイプラインにするには、まずアプリケーションからめなければなりません、のにするはありません。 `--stage STAGE` しない、CLIはなでしますが、デフォルトをきすることもできます。パイプラインのはアプリからもされますが、コマンドラインに `NAME` をするか、プロンプトがされたらのをすることでオーバーライドできます。

```
heroku pipelines:create -a example
```

する

アプリは、のステージでにされます

```
heroku pipelines:promote -r staging
```

のアプリまたはのアプリにプロモートすることもです

```
heroku pipelines:promote -r staging --to my-production-app1,my-production-app2
```

ヘルプコマンド

のをむパイプラインコマンドのなりリストは、コンソールでできます

```
heroku help pipelines
```

オンラインでパイプラインをむ <https://riptutorial.com/ja/heroku/topic/2389/パイプライン>

8: ビルドパック

Examples

ビルドパックの

HerokuはRuby、Node.js、Clojure、Python、Java、Gradle、Grails、Scala、Play、PHP、Goのビルドパックをにサポートしています。

ビルドパックはHerokuによってにされますが、CLIをしてのもですることもできます。

1. アプリ

```
heroku create <app_name> --buildpack <buildpack_name>
```

2. で、

```
heroku buildpacks:set <buildpack_name>
```

ビルドパックはまたはURLのいずれかをしてできます。 PHPビルドパックののに、

```
heroku buildpacks:set heroku/php
```

または

```
heroku buildpacks:set https://elements.heroku.com/buildpacks/heroku/heroku-buildpack-php
```

のビルドパック

アプリケーションにはのビルドパックをめることもできます。それは`add`をってすることができます

```
heroku buildpacks:add --index 1 <buildpack_name>
```

ここで、`--index`パラメータはbuildpackのをします。

いう、

```
heroku buildpacks:set heroku/php
heroku buildpacks:add --index 1 heroku/nodejs
```

buildpackのをのようにします。

えておいてください Herokuアプリには1つのパブリックポートしかありません。したがって、どちらかのポートでサービスをします。 `procfile`、 `procfile`が `web: node server.js` でされている、ノードアプリケーションはポート80、それのはPHPでされます。ただし、ビルドはされたでされます。のアプリケーションがなは、のプロジェクトをし、いにできるようにします。

オンラインでビルドパックをむ <https://riptutorial.com/ja/heroku/topic/6126/ビルドパック>

9: ヘロクの

Examples

Heroku プラットフォームのすべてののリスト

1. ログデフォルトでは、Herokuは1500のログのみをします。ログの1500のがなは、1がするため
にっている **アドオン** Herokuのをします。
2. ルータ HTTPにはのために30のタイムアウトがあり、そのは55のタイムアウトがあります。に
は1MBのバッファができます。
3. **Dynos**した **タイプ** についたDyno メモリの。フリーダイノスの、30のにるにがせられます。さ
らに、されたアカウントにはの1000 dynoのプールがあり、されていないアカウントには550が
りてられます。アプリケーションには100 dynosがあり、プロセスタイプは10 dynosにすることは
できません。フリーdynoタイプは、2つのダイノーをつことができます。
4. **Config Vars** アプリの **キー** との **ペア** は32kbにされています。
5. ビルドユーザーは、アプリケーションごと、ユーザーごとにHeroku Git reposへの75のリクエ
ストにされています。チェックアウトのサイズは1GBをえることはできません。スラッグのサイ
ズは300 MBにされ、コンパイルのさは15をえることはできません。
6. データクリップすべてのクエリは10まででき、100,000をすことができます。
7. **Heroku Postgres** は、ごとに4から15までのなるによって **なります**。
8. **API** Heroku APIへのびしは2400 /にされています。
9. メンバーシップのエンタープライズアカウントの、500のメンバーとのメンバー、25のメンバ
ーがされます。
10. アプリケーションされたユーザーが100のアプリをできます。のユーザーは5つのアプリケー
ションにされています。

オンラインでヘロクのをむ <https://riptutorial.com/ja/heroku/topic/6190/ヘロクの>

10: ログ

- `$ heroku ログ`
- `$ heroku logs -n 200`
- `$ heroku logs --tail`
- `$ heroku ログ--dynoルーター`
- `$ heroku logs --source app`
- `$ heroku logs --source app --dyno worker`
- `$ herokuのログ--source app --tail`

Examples

ログの

Herokuは、あなたのアプリケーションのログの3つのカテゴリをします

- **アプリケーションログ** - アプリケーションからされます。これには、アプリケーション、アプリケーションサーバー、およびライブラリからされたログが含まれます。 フィルタ `--source app`
- **システムログ** - アプリケーションにわってHerokuプラットフォームインフラストラクチャがしたアクションクラッシュしたプロセスの、Webダイノのスリープやスリープ、アプリケーションのによるエラーページのなどにするメッセージ。 フィルター `--source heroku`
- **APIログ** - 新しいコードのデプロイ、プロセスのスケーリング、メンテナンスモードのりえなど、あなたやのがあなたのアプリケーションでするアクションにするメッセージです。 フィルター `--source heroku --dyno api`

ログ

ログのはのとおりです。

```
timestamp source[dyno]: message
```

- **Timestamp** - ログラインがdynoまたはコンポーネントによってされたときにされた。タイムスタンプはRFC5424でされているフォーマットであり、マイクロのが含まれています。
- **ソース** - あなたのアプリのすべてのdynosWeb dynos、バックグラウンドワーカー、cronにソース、 `app` ます。 HerokuのすべてのシステムコンポーネントHTTPルーター、 `heroku` マネージャーには、ソース `heroku` ます。
- **Dyno** - ログラインをきんだdynoまたはコンポーネントの。たとえば、 `worker3`は `worker.3`とされ、Heroku HTTPルーターはルーターとしてされ `router`。

- メッセージ - ログの。 dynosによってされた10000バイトをえるは、ななしで10000バイトのチャンクにされます。チャンクは々のログラインとしてされます。

ログをする

ログをするには、 `heroku logs` コマンドをします。

```
$ heroku logs
```

`logs` コマンドは、デフォルトで100のログをします。 `--num` または `-n` オプションをすると、するログの数を1,500まで増やせます。

```
$ heroku logs -n 200
```

リアルタイムテール

`tail -f` とに、リアルタイムテールはのログをし、リアルタイムログをするためにセッションをいたままにします。アプリからログのライブストリームをすることで、ライブアプリケーションのログをすることができます。のデバッグに役立ちます。 `--tail` または `-t` をしてログをテールすることができます。

```
$ heroku logs --tail
```

したら、**Ctrl + C**をしてプロンプトに戻ります。

ログフィルタリング

あなたがのソース、のダイノ、またはそのでログをしたいは、することができます `--source` または `-s` と `--dyno` または `-d` フィルタリングの

```
$ heroku logs --dyno router
2012-02-07T09:43:06.123456+00:00 heroku[router]: at=info method=GET path="/stylesheets/dev-center/library.css" host=devcenter.heroku.com fwd="204.204.204.204" dyno=web.5 connect=1ms service=18ms status=200 bytes=13
2012-02-07T09:43:06.123456+00:00 heroku[router]: at=info method=GET path="/articles/bundler" host=devcenter.heroku.com fwd="204.204.204.204" dyno=web.6 connect=1ms service=18ms status=200 bytes=20375
```

```
$ heroku logs --source app
2012-02-07T09:45:47.123456+00:00 app[web.1]: Rendered shared/_search.html.erb (1.0ms)
2012-02-07T09:45:47.123456+00:00 app[web.1]: Completed 200 OK in 83ms (Views: 48.7ms | ActiveRecord: 32.2ms)
2012-02-07T09:45:47.123456+00:00 app[worker.1]: [Worker(host:465cf64e-61c8-46d3-b480-362bfd4ecff9 pid:1)] 1 jobs processed at 23.0330 j/s, 0 failed ...
2012-02-07T09:46:01.123456+00:00 app[web.6]: Started GET "/articles/buildpacks" for 4.1.81.209 at 2012-02-07 09:46:01 +0000
```

```
$ heroku logs --source app --dyno worker
```

```
2012-02-07T09:47:59.123456+00:00 app[worker.1]: [Worker(host:260cf64e-61c8-46d3-b480-362bfd4ecff9 pid:1)] Article#record_view_without_delay completed after 0.0221
2012-02-07T09:47:59.123456+00:00 app[worker.1]: [Worker(host:260cf64e-61c8-46d3-b480-362bfd4ecff9 pid:1)] 5 jobs processed at 31.6842 j/s, 0 failed ...
```

また、フィルタリングスイッチを`--tail`とみわけて、フィルタリングされたのリアルタイムストリームをすることもできます。

```
$ heroku logs --source app --tail
```

オンラインでログをむ <https://riptutorial.com/ja/heroku/topic/8327/ログ>

11:

- ""{...}

Examples

ポーワー

bowerとそのコンポーネントをにインストールするには、

1. package.json bowerをする

```
"dependencies": {  
  "bower": "^1.7.9"  
}
```

2. scripts をして postinstall コマンドをする

```
"scripts": {  
  "postinstall": "./node_modules/bower/bin/bower install"  
}
```

3. bower_components をインストールするディレクトリをする .bowerrc ファイルをします。それは、bower_components がルートディレクトリにインストールされます。

```
{  
  "directory" : "app/bower_components"  
}
```

、Heroku は `npm install` に `bower install` コマンドをします

オンラインでをむ <https://riptutorial.com/ja/heroku/topic/6665/>

12:

- `git push heroku master`

Examples

Gitでデプロイする

gitであなたのアプリをする

あなたがHerokuにアプリをプッシュするに、ローカルのGitリポジトリをし、それにファイルをコミットするがあります。たとえば、ディレクトリmyappにアプリケーションがある、そのディレクトリにしいリポジトリをします。

```
$ cd myapp
$ git init
Initialized empty Git repository in .git/
$ git add .
$ git commit -m "my first commit"
Created initial commit 5df2d09: my first commit
 44 files changed, 8393 insertions(+), 0 deletions(-)
 create mode 100644 README
 create mode 100644 Procfile
 create mode 100644 app/controllers/source_file
...
```

これはローカルリポジトリで、`.git`ディレクトリにします。まだどこにももられていない。あなたはリモートをし、Herokuにコードをデプロイするためのプッシュをうがあります。

Heroku リモートをする

```
$ heroku create
Creating falling-wind-1624... done, stack is cedar-14
http://falling-wind-1624.herokuapp.com/ | https://git.heroku.com/falling-wind-1624.git
Git remote heroku added
```

のアプリケーションをつGitリポジトリ`heroku git:remote`コマンドは、あなたのアプリケーションのgit urlについてこのリモートをします。

```
$ heroku git:remote -a falling-wind-1624
Git remote heroku added.
```

コードの

プッシュのリモートブランチをするがあります。あなたはのプッシュをすることができます

```
$ git push heroku master
Initializing repository, done.
updating 'refs/heads/master'
...
```

masterのブランチをプッシュするには、のをします。

```
$ git push heroku yourbranch:master
```

オンラインでをむ <https://riptutorial.com/ja/heroku/topic/8325/>

クレジット

S. No		Contributors
1	ヒロクをめよう	Community , rdegges , thejonanshow
2	Heroku node.js Hello World	Johan Hoeksma
3	Heroku Postgres	Denis Savchuk , Hardik Kanjariya ツ , Thamilan
4	Herokuアドオン	jophab
5	Herokuエラーコード	Sender
6	コマンドライン	Sender
7	パイプライン	Thamilan
8	ビルドパック	Thamilan
9	へロクの	autoboxer , Thamilan
10	ログ	Sender
11		Thamilan
12		Sender