クラウド(PaaS)体験

～はじめてのRuby製ウェブアプリケーション開発～

Ruby製ウェブアプリケーションの開発支援環境と実行環境を提供するサービスのクラウド（PaaS）を利用して、カウンタ付きのホームページを作成します。
本文書の対象読者は学生（高校生以上）、新社会人といったプログラミングの初心者を想定しています。Rubyでのプログラミングの経験があることも想定しています。なお、経験者には物足りない内容になっています。

目次

- 第１章 クラウド(PaaS)
  - クラウドの種類や本文書で扱うクラウドサービスHerokuについて説明します。（15分）
- 第２章 クラウドの利用準備
  - クラウドを利用するためのコンピュータの設定方法やHerokuのサインアップ手順を説明します。（30分）
- 第３章 クラウドのアプリ開発
  - Heroku上で動作するWebアプリケーションを開発します。開発するのは閲覧数をカウントするカウンタが付いたホームページです。（2時間）

第１章　クラウド(PaaS)

- IaaS：(ア)イアース
  - Infrastructure as a Service
  - インフラ

- PaaS：パース
  - Platform as a Service
  - (アプリケーション)プラットフォーム

- SaaS：サース
  - Software as a Service
  - ソフトウェア

PaaS

Heroku: Ruby,Java,Python,Clojure,Scala,Node.jsなど
　http://www.heroku.com/
　読み方：へぇろぉーく
　料金：無料/有料
　特徴：
　　Rubyのサポートが充実
　　スケールアウト可能
　　正式サービス

Google App Engine: Python,Java
　https://developers.google.com/appengine/?hl=ja

Cloud Foundry: Java,Ruby,node.js,Groovy,Scalaなど
　http://cloudfoundry.com/

Mogok: Ruby
　http://mogok.jp/

Squale: Ruby
　http://sqale.jp/

第２章　クラウドの利用準備

- インターネットへの接続
  - メールアドレスとメールを受信できる環境が必要
- エディタ
  - Windows
    - TeraPad(http://www5f.biglobe.ne.jp/~t-susumu/library/tpad.html)
  - Mac
    - CotEditor(http://sourceforge.jp/projects/coteditor/)
- Ruby on Railsのインストール
  - Windows 7またはMac OS X 10.7
  - Ruby 1.9.3以降
  - Ruby on Rails 3.2以降
  - SQLite3
  - ※RailsInstaller(http://railsinstaller.org)をインストールすればOK

TeraPadを使う場合、ファイルを開くときに「ファイルの種類」として「すべて」を指定してください。初期状態ではファイルを開くときにRubyのプログラムを選択できません。

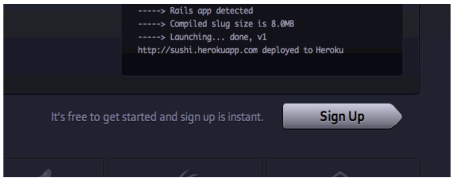
Heroku関連ツールのインストール

- コマンドプロンプトを起動
  - 以降、Macの場合はターミナルに読み替える
- 次のコマンドを順に実行
  - (環境によっては 「rvm use 1.9.3-p194」が必要)
  - gem install heroku --no-ri --no-rdoc
  - gem install foreman --no-ri --no-rdoc
- コマンドプロンプトを閉じる

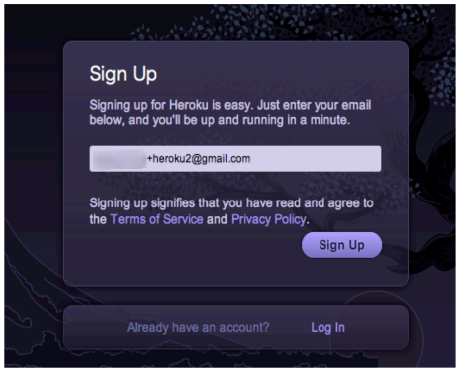
Herokuのサインアップ

- 1.ブラウザで http://www.heroku.com にアクセス
- 2.[Sign Up]ボタンをクリック
- 3.すぐにメール受信可能なメールアドレスを入力
- 4.Herokuからのメールを待つ
- 5.メールが届いたらそれに含まれるURLをクリック
- 6.Herokuで使うパスワードを入力
- 7.[Save]ボタンをクリック

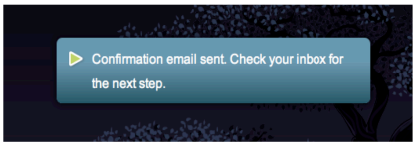
1.
2.



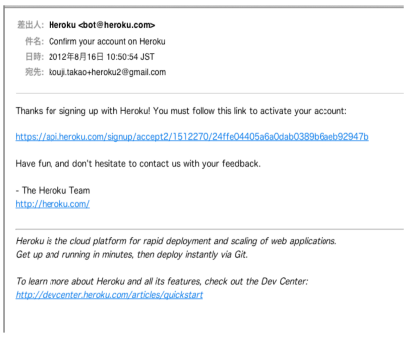
3.



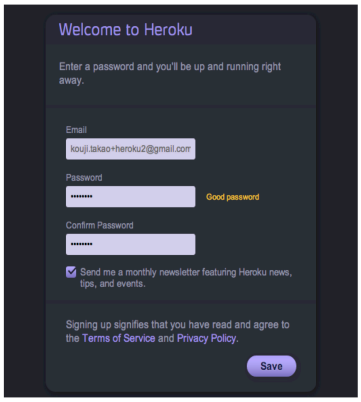
4.



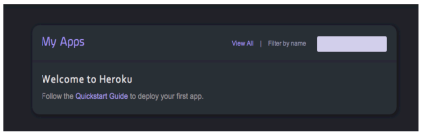
5.



6.



7.



第３章　クラウドのアプリ開発

- 開発するWebアプリケーション
  - カウンタ付きホームページ

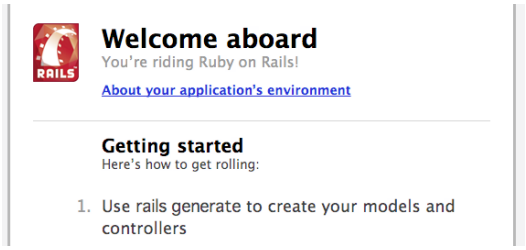
開発するWebアプリケーションはカウンタ付きホームページです。

具体的には次の2つのページを表示できるようにします。
・トップページ
　簡単なあいさつ、カウンタ、アクセスした日時を表示
・連絡先
　メールアドレスを表示

Webアプリケーションを配置したサーバのURLがhttp://www.example.comだとすると、http://www.example.com/ または http://www.exmaple.com/home/index でトップページを表示できるようにします。
また、http://www.example.com/home/contactで連絡先ページを表示できるようにします。

はじめてのRailsアプリ

- コマンドプロンプトで次のコマンドを順番に実行
  - rails new hello
  - cd hello
  - rake db:create
  - rails s
    ※後ほど別のコマンドプロンプトを起動するため、以降はこちらをコマンドプロンプトAと記述
- ブラウザで次のURLにアクセス
  - http://localhost:3000/



Windows 7の場合、「rails s」コマンドを実行後、セキュリティ警告が表示されることがあります。

トップ・連絡先ページのひな形の作成

- 別のコマンドプロンプトを起動
  - 以降はコマンドプロンプトBと記述
- 次のコマンドを順番に実行
  - cd hello
  - rails generate controller Home index contact
- ブラウザで次のURLにアクセス
  - http://localhost:3000/home/index

- ブラウザで次のURLにアクセス
  - http://localhost:3000/home/contact

デプロイの準備

- hello/Gemfileを修正
- hello/config/application.rbを修正
- コマンドプロンプトBで次のコマンドを実行
  - bundle install
  - git init
  - git add .
  - git commit -m 'updated'

helloフォルダのGemfile(以降は「hello/Gemfile」と記述)を以下のように修正
<修正前(該当箇所のみ記述)>
```
gem 'sqlite3'
```
<修正後(該当箇所のみ記述)>
```
group :development, :test do
  gem 'sqlite3'
end
group :production do
  gem 'pg'
end
```

hello/config/application.rbを以下のように修正
<修正前(該当箇所のみ記述)>
```
# config.time_zone = 'Central Time (US & Canada)'
```
<修正後(該当箇所のみ記述)>
```
config.time_zone = 'Tokyo'
```

コマンドプロンプトBで次のコマンドを実行
```
bundle install
git init
git add .
git commit -m 'updated.'
```

はじめてのデプロイ

- コマンドプロンプトBで次のコマンドを実行
  - heroku login
    - (Windowsの場合は以下を実行)
    - heroku keys:add ¥Users¥□¥.ssh¥id_rsa.pub
  - heroku create
    - ここで表示される「○○.herokuapp.com」を記録
  - git push heroku master
  - heroku open
- ブラウザで次のURLにアクセス
  - http://○○.herokuapp.com/home/index
  - http://○○.herokuapp.com/home/contact

各コマンドの実行例

heroku login
(Windows 7でRailsInstallerを利用する場合はSSHの鍵はC:\Users\□/.ssh/id_rsa.pub)
Enter your Heroku credentials.
Email: ○○@△△ (サインアップ時のメールアドレスの入力)
Password (typing will be hidden): (サインアップ時のパスフレーズの入力)
Found the following SSH public keys:
1) id_rsa.pub
Which would you like to use with your Heroku account? 1 (SSHの公開鍵の選択)
Uploading SSH public key /Users/□/.ssh/id_rsa.pub... done
Authentication successful.

heroku create
(「salty-castle-6738」に対応する部分を記録しておく)
Creating salty-castle-6738... done, stack is cedar
http://salty-castle-6738.herokuapp.com/ | git@heroku.com:salty-castle-6738.git
Git remote heroku added

git push heroku master
(省略)
http://salty-castle-6738.herokuapp.com deployed to Heroku

トップ・連絡先ページの改良

- hello/app/views/home/index.html.erbの修正

```
◯◯のホームページ
ようこそ、◯◯のホームページへいらっしゃいました。
```

- hello/app/views/home/contact.html.erbの修正

```
連絡先
メールアドレスは <a href="mailto:△△">△△</a> です。
```

- コマンドプロンプトAでCtrl+Cを押してから、次のコマンドを実行
  - rails s
- ブラウザで以下のURLにアクセス
  - http://localhost:3000/home/index
  - http://localhost:3000/home/contact

ファイルの文字コードはUTF-8、改行文字はCR+LFで保存すること

hello/app/views/home/index.html.erbの修正後
(◯◯には自分の名前を入れること)

```
<h1>◯◯のホームページ</h1>
<p>ようこそ、◯◯のホームページへいらっしゃいました。</p>
```

hello/app/views/home/contact.html.erbの修正後
(△△にはHerokuにサインアップしたメールアドレスを入れること)

```
<h1>連絡先</h1>
<p>メールアドレスは <a href="mailto:△△">△△</a> です。</p>
```

＜課題＞上記の2つのファイルにはHTMLのタグを自由に記述することができます。<hr>や<h1〜4>、style属性による色付けなどを施してホームページらしくしましょう。

サイトマップ・日時表示の追加

- hello/app/views/layouts/application.html.erbの修正

サイトマップ

・トップ
・連絡先

- hello/app/views/home/index.htmlの修正

○○のホームページ

ようこそ、○○のホームページへいらっしゃいました。
現在の時刻は □年□月□日 □時□分□秒 です。

- ブラウザで以下のURLにアクセス
  - http://localhost:3000/home/index
  - http://localhost:3000/home/contact

hello/app/views/layouts/application.html.erbの修正後
(修正箇所周辺のみを記述。修正箇所は太字部分のみ。)

```
  <%= yield %>

  <hr />
  <h3>サイトマップ</h3>
  <ul>
    <li><%= link_to("トップ", home_index_path) %></li>
    <li><%= link_to("連絡先", home_contact_path) %></li>
  </ul>

  </body>
  </html>
```

hello/app/views/home/index.html.erbの修正後
(下部に以下を追記。)

```
  <p>現在の時刻は<%= Time.zone.now.strftime("%Y年%m月%d日 %H時%M分%S秒) %> です。</p>
```

＜課題＞hello/app/views/layouts/application.html.erbはHTMLのタグを自由に記述することができます。<hr>や<h1～4>、style属性による色付けなどを施してホームページらしくしましょう。ただし、「<%= ○○ %>」の部分は変更しないでください。

再デプロイ

- コマンドプロンプトBで次のコマンドを順に実行
  - git add .
  - git commit -m 'updated'
  - git push heroku master

- ブラウザで以下のURLにアクセス
  (◯◯の箇所はheroku createのときに記録したもの)
  - http://◯◯.herokuapp.com/home/index
  - http://◯◯.herokuapp.com/home/contact

再度デプロイするときは、heroku loginやheroku createは必要ありません。

カウンタの追加(1)

- コマンドプロンプトBで次のコマンドを順に実行
  - rails generate model counter count:integer
  - rails generate migration InsertIntoCounters
- hello/db/migrate/<日時>_insert_into_counters.rbを修正
- コマンドプロンプトBで次のコマンドを実行
  - rake db:migrate
- hello/app/views/home/index.html.erbを修正

> ○○のホームページ
>
> ようこそ、○○のホームページへいらっしゃいました。
> あなたは △ 番目の訪問者です。
> 現在の時刻は □年□月□日 □時□分□秒 です。

- ブラウザで以下のURLにアクセス
  - http://localhost:3000/home/index

hello/db/migrate/<日時>_insert_into_counters.rbの修正後
(修正箇所は太字部分のみ)

```
class InsertIntoCounters < ActiveRecord::Migration
  def up
    Counter.create(count: 0)
  end

  def down
    Counter.delete_all
  end
end
```

hello/app/views/home/index.html.erbの修正後
(修正箇所周辺のみ記述。修正箇所は太字のみ)

```
<%
    @counter = Counter.first
    Counter.increment_counter(:count, @counter.id)
    @counter.reload
%>
<h1>○○のホームページ</h1>
<p>ようこそ、○○のホームページへいらっしゃいました。</p>
<p>あなたは <%= @counter.count %> 番目の訪問者です。</p>
```

カウンタの追加(2)

- hello/app/views/home/index.html.erbを修正
- hello/app/controllers/home_controller.rbを修正
- ブラウザで以下のURLにアクセス
  - http://localhost:3000/home/index

hello/app/views/home/index.html.erbの修正後
(修正箇所周辺のみ記述。修正箇所は太字のみ)

```
  ここにあった「<%」から「%>」まで削除
  <h1>○○のホームページ</h1>
```

hello/app/controllers/home_controller.rbの修正後
(修正箇所周辺のみ記述。修正箇所は太字のみ)

```
 def index
   @counter = Counter.first
   Counter.increment_counter(:count, @counter.id)
   @counter.reload
 end
```

/でのアクセス

- hello/config/routes.rbを修正
- hello/public/index.htmlを削除
  - コマンドプロンプトBで次のコマンドを実行
    - cd public
    - rm index.html
    - git rm index.html
    - cd ..
- ブラウザで以下のURLにアクセス
  - http://localhost:3000/

hello/config/routes.rbの修正後
<修正前(該当箇所のみ記述)>
```
  # root :to => 'welcome#index'
```
<修正後(該当箇所のみ記述)>
```
  root :to => 'home#index'
```

DBの更新を含むデプロイ

- コマンドプロンプトBで次のコマンドを順に実行
  - git add .
  - git commit -m 'updated'
  - git push heroku master
  - heroku run:detached rake db:migrate
- ブラウザで以下のURLにアクセス
  - http://◯◯.herokuapp.com/

アプリ名の変更・アプリの削除

- アプリ名の変更
  - コマンドプロンプトBで次のコマンドを実行
    - heroku apps:rename 新しいアプリ名
  - ブラウザで以下のURLにアクセス
    - http://新しいアプリ名.herokuapp.com/
- アプリの削除
  - コマンドプロンプトBで次のコマンドを実行
    - heroku apps:destroy アプリ名
- アプリの再作成（間違って削除したときなど）
  - コマンドプロンプトBで次のコマンドを実行
    - heroku create --app アプリ名
    - git push heroku master
    - heroku run:detached rake db:migrate

heroku apps:destroyの実行例

```
heroku apps:destroy アプリ名
 !    WARNING: Potentially Destructive Action
 !    This command will destroy takao-kouji-web (including all add-ons).
 !    To proceed, type "アプリ名" or re-run this command with --confirm  アプリ名
> アプリ名 (ここにアプリ名を入力)
Destroying アプリ名 (including all add-ons)... done
```

まとめ

- 第１章 クラウド(PaaS)
  - クラウドの種類や本文書で扱うクラウドサービスHerokuについて説明しました。
- 第２章 クラウドの利用準備
  - クラウドを利用するためのコンピュータの設定方法やHerokuのサインアップ手順を説明しました。
- 第３章 クラウドのアプリ開発
  - Heroku上で動作するWebアプリケーションを開発しました。

以上です。
おつかれさまでした。