

## 【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」


1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分


### ■ 2-0：講座を受ける前に

#### 【ZOOMの使い方】

##### リアクションについて

ZOOM画面下部に「反応」というボタンがありますので、

- ・講師が皆さんの進捗を伺いますので、何も問題ない場合には、右側のいいねマーク 

- ・何かわからない点・つまづいた点がありましたら、左側の拍手マーク を押してください。

##### チャット機能について

わからないことがあれば、基本的にチャットで質問してください。

##### チャット機能で改行する方法

Windows → Shift + Enter.

Mac → control + Return (Enter)

##### 質問の仕方について

下記のように、質問内容・入力したコマンドの行全体・出力結果をチャットしてくださると助かります。

ex)

下記のエラーが出ます。どうすればいいですか？

```
ec2-user:~/environment/contact_app/techgym_rails_course02 (lesson4) $ git checkout lesson4
```

```
Already on 'lesson4'
```

```
Your branch is up-to-date with 'origin/lesson4'.
```

##### ミュートについて

基本的にはミュートしててください。

チャットでは、質問しづらい内容がありましたら、ミュートを解除し、発言してください。

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

サンプルソースの公開場所：[https://github.com/techgymjp/techgym\\_rails\\_course02](https://github.com/techgymjp/techgym_rails_course02)

☆ 実行環境はCloud9(<https://aws.amazon.com/jp/cloud9/>)を使用する。

☆ 対象のgithubリポジトリをクローンする。

■ 2-0-1：実行環境を整えよう

【手順】

環境設定として下記のコマンド入力してください。

コマンドは、Terminalに下記図のように入力し、EnterキーまたはReturnキーを押してください。

※ \$マークは、すでに入力されているため、\$より後ろを入力してください。

コマンドを実行しても、何も表示されない場合がありますが、問題ありません。

```
ec2-user:~/environment $ mkdir techgym_rails
```

techgym\_railsという名前のフォルダを作成する。

```
$ mkdir techgym_rails
```

techgym\_railsフォルダに移動する。

```
$ cd techgym_rails
```

対象のgithubリポジトリをクローンする。

クローン： github上のプロジェクトをカレントディレクトリに複製する。

```
$ git clone https://github.com/techgymjp/techgym\_rails\_course02.git
```

techgym\_rails\_course02フォルダに移動する。

```
$ cd techgym_rails_course02
```

プロジェクトに必要なプログラムをインストールする。

```
$ bundle install --path vendor/bundle
```

※ postgresqlがエラーが発生した場合

```
An error occurred while installing pg (1.2.3), and Bundler cannot continue.  
Make sure that `gem install pg -v '1.2.3' --source 'https://rubygems.org/'` succeeds before bundling.
```

必要なパッケージをインストールする。

```
$ sudo yum install postgresql postgresql-server postgresql-devel postgresql-contrib -y
```

データベースの初期化

```
$ sudo service postgresql initdb
```

```
$ bundle install --path vendor/bundle
```

データベースサーバーの起動

```
$ sudo service postgresql start
```

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

データベースをセットアップする。

\$ bundle exec rake db:setup

※ データベース作成時にpostgresqlのエラーが発生した場合

```
FATAL: role "ec2-user" does not exist
Couldn't create 'contact_app_development' database. Please check your configuration.
rake aborted!
```

ユーザーの作成

\$ sudo -u postgres createuser -s ec2-user

\$ bundle exec rake db:setup

※ 下記のエラーが発生した場合

Could not find public\_suffix-4.0.4 in any of the sources

Run `bundle install` to install missing gems.

\$ bundle install --path vendor/bundle

\$ bundle exec rake db:setup

作業ブランチを切り替える

\$ git checkout -b lesson5 remotes/origin/lesson5

Railsのサーバーを起動する。

\$ bundle exec rails server

【実行結果】

URL: /

ex) <https://f24e3029423e4xxxxxx38c8888d4.vfs.cloud9.ap-northeast-1.amazonaws.com/>

ページの表示方法がわからない方は、15ページ「Cloud9でブラウザを立ち上げる」をご確認ください。

## 案件管理

ID	タイプ	タイトル	報酬(最小)	報酬(最大)	ステータス
----	-----	------	--------	--------	-------

新規作成

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

サンプルソースの公開場所：[https://github.com/techgymjp/techgym\\_rails\\_course02](https://github.com/techgymjp/techgym_rails_course02)

☆ 実行環境はCloud9(<https://aws.amazon.com/jp/cloud9/>)を使用する。

☆ 対象のgithubリポジトリをクローンする。

■ 2-0-2：実行環境を整えよう(既に環境設定が終了している方)

【手順】

環境設定として下記のコマンド入力してください。

DBサーバーの起動

```
$ sudo service postgresql start
```

techgym\_railsフォルダに移動する。

```
$ cd techgym_rails
```

techgym\_rails\_course01フォルダに移動する。

```
$ cd techgym_rails_course02
```

新しいブランチをダウンロードする。

```
$ git fetch origin
```

前回の修正が残っていた場合の対処(修正が残ってなくても、実行して問題ありません。)

```
$ git add .
```

```
$ git commit -m "保存機能追加"
```

```
$ git checkout lesson4
```

```
$ git branch -D lesson5
```

```
$ git checkout -b lesson5 remotes/origin/lesson5
```

プロジェクトに必要なプログラムをインストールする。

```
$ bundle install --path vendor/bundle
```

データベースの修正を反映する

```
$ bundle exec rails db:migrate
```

Railsのサーバーを起動する。

```
$ bundle exec rails server
```

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

■ 2-5 : 新しいタスクを作成しよう : lesson5

【問題】

スクレイピングするためのタスクを新たに用意し、事前に用意しているモジュールから関数を呼び出そう。

【修正する内容】

ファイル : lib/tasks/scraping.rake

修正する内容 : 新たなタスクの追加(task fetch\_lancers)

モジュールの読み込み(lib/scraping\_work\_lancers.rb)

モジュール内の関数の呼び出し(sample\_function)

【実行結果】

タスクが正常に設定されているか確認

\$ bundle exec rake -T scraping

→ rake scraping:fetch\_crowdworks # クラウドワークスの情報取得

rake scraping:fetch\_lancers # ランサーズの情報取

タスクの実行

\$ bundle exec rake scraping:fetch\_lancers

→ "ランサーズモジュールが正しく読み込まれています"

【ヒント】

- タスクfetch\_crowdworksを参考に新しいタスクfetch\_lancersを作成しましょう。
- モジュールScrapingWork同様、ScrapingWorkLancersモジュールも用意しています。
- 一つのnamespace内に複数のタスク作成することができます

## 【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

### ■ 2 - 6 : 案件募集が終了しているかどうかを取得しよう : lesson6

#### 【はじめに】

```
$ git add .  
$ git commit -m "ランサーズ用タスク追加"  
$ git checkout -b lesson6 remotes/origin/lesson6
```

#### 【問題】

募集期限が終了している案件には、「この募集は終了しました。」と表示されるので、表示されている文言を元に案件募集が終了しているかどうかを判定しましょう。

取得するページの例)

<https://www.lancers.jp/work/detail/2908104>

#### 【修正する内容】

ファイル : lib/tasks/scraping.rake

修正するタスク : fetch\_lancers

修正する内容 : URLを指定して、対象ページのHTMLを取得し、doc変数に格納しよう

ScrapingWorkLancersモジュールのis\_finish関数を呼び出し返された値を出力しましょう

ファイル : lib/scraping\_work\_lancers.rb

修正する関数 : is\_finish

修正する内容 : 対象ページの特定の要素に終了という文字列が含まれているかどうかの真偽値を返す

#### 【実行結果】

タスクの実行

```
$ bundle exec rake scraping:fetch_lancers
```

```
→ "ランサーズモジュールが正しく読み込まれています."  
true
```

#### 【ヒント】

□ Nokogiriオブジェクト atメソッドは、対象の要素が存在しない時には、nilを返します。

□ Rails上オブジェクトでは有効な値かどうか判別するために、真偽値を返すpresent?メソッド用意されています。

例) status = nil

```
p status.present?
```

```
→ false
```

□ 文字列オブジェクトには、include?メソッドが定義されています。

include?メソッドは、引数に文字列をとり、引数の文字列が含まれるかどうかを真偽値で返します。

例) p "abcde".include?("bcd")

```
→ true
```

□ 複数の真偽値を論理演算し、一つの真偽値を返すことができます。

例) p true && true → true

```
p true && false → false
```

```
p false && false → false
```

## 【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

### ■ 2-7：詳細を取得しよう：lesson7

#### 【はじめに】

```
$ git add .
```

```
$ git commit -m "案件募集が終了しているかどうかを取得"
```

```
$ git checkout -b lesson7 remotes/origin/lesson7
```

#### 【問題】

案件の詳細を取得してください。また、詳細は複数のタグに別れていますので全て取得した上で文を結合する必要があります。

#### 【修正する内容】

ファイル：lib/tasks/scraping.rake

修正するタスク：fetch\_lancers

実行する内容：ScrapingWorkLancers内のdetailメソッドを呼び出し、返された値を表示する

ファイル：lib/scraping\_work\_lancers.rb

修正する関数：detail

修正する内容：複数のタグに別れている詳細を取得・結合し、値を返す

#### 【実行結果】

タスクの実行

```
$ bundle exec rake scraping:fetch_lancers
```

```
"ランサーズモジュールが正しく読み込まれています。"
true
"依頼のジャンル\n                IT・通信・インターネット依頼の目的・背景\n表示される広告のURL他を取得し、MySQLに格納するツールの作成をお願いします。\\r\\n社内の技術者がRub
レイビング周りのスキル\\r\\n ・コストを考えたインフラ構築\\r\\n ・bot判定されることを想定し、ボットを回避
えた際の対策(負荷分散、並列処理)\\r\\n・Ruby on Rails\\r\\n・システム設計、及びDB設計\\r\\n・パッチ
\\n・取得しないURL(ドメイン)の設定\\r\\n・テストコード\\r\\n・エラー検知\\r\\n・バッチ処理の作成\\r\\n\\r\\n
かいUIの作り込みは弊社内で行うので不要です。フロント側のUIはAdobe XDで用意しています。(画像でのお
定ですが、弊社内で行うため制作いただく必要はありません。\\r\\n\\r\\n■制作の流れ\\r\\n・システム設計を行
通知を行ってまいります\\r\\n・弊社管理のAWSに構築をお願いします。\\r\\n・AWS上にクラウドジョブを作ってい
ます\\r\\n\\r\\n■コミュニケーションについて\\r\\n相対する私はWebサービスでのプロダクトマネージャー経験が
ョンを取ることは可能です。\\r\\nまた専用のSlackにご招待しますので、そちらでやりとりをさせていただきます。\\r\\
Nを対策する方法やアイデア(実装時に変更することは構いません)\\r\\n・納品後のメンテナンス、及び追加機能
分らないので、相談して決めさせていただければと思います。希望開発言
\n                Ruby on Rails補足説明\n設定なし分野\n相談して決めさせていただければと思います。用意してあるもの\n書"
```

#### 【ヒント】

- Nokogiriオブジェクトから複数のタグを取得したい時には、atメソッドではなく、searchメソッドを使用します。  
searchメソッドはXPathを引数にとり、XPathに対応したオブジェクトの配列を返します。
- 配列の全ての要素に対して、特定のメソッドを実行させたい時には、mapメソッドを使用します。  
例) array.map { |dl| dl.text }
- 文字列の配列を結合して、一つの文字列を作成する際にはjoinメソッドを使用します。

【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

■ 2 - 8 : 報酬を取得し、案件を保存しよう : lesson8

【はじめに】

```
$ git add .
```

```
$ git commit -m "詳細追加"
```

```
$ git checkout -b lesson8 remotes/origin/lesson8
```

【問題】

報酬(最小・最大)を取得しよう。ただし、報酬に関する方法は、headタグ内のmetaタグ(description)に記載されています。

また、取得した全て情報をデータベースに保存してください。

【修正する内容】

ファイル : lib/tasks/scraping.rake

修正するタスク : fetch\_lancers

修正する内容 : 全ての情報を取得し、データベースに保存する

ファイル : lib/scraping\_work.rb

修正する関数 : reward

修正する内容 : 報酬(最小・最大)を取得して、正しい値であれば、最小値と最大値を返す。不正な値であれば、nilを返す

【実行結果】

タスクの実行

```
$ bundle exec rake scraping:fetch_lancers
```

案件管理画面にアクセス(URL: /)

ex) <https://f24e3029423e4xxxxxx38c8888d4.vfs.cloud9.ap-northeast-1.amazonaws.com/>

## 案件管理

ID	タイプ	タイトル	報酬(最小)	報酬(最大)	ステータス			
1	ランサーズ	Ruby on Rails を使っ...	5000	6000	終了	表示	編集	削除
新規作成								



【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

【ヒント】

- 全ての情報を取得する時には、ScrapingWorkLancersモジュールのfetch\_workメソッドを利用しましょう。fetch\_workメソッドは、第一引数に、URL(文字列)、第二引数にdoc(Nokogiriオブジェクト)をとります。
- 特定要素のNokogiriオブジェクトから属性を指定して、属性値を取得したい場合には、下記のように記述する。  
例) doc.at(/XPath/)[:content]
- 文字列からある一定の法則に基づいて、一部の文字列を取得指定場合にはよく正規表現が使われる。  
例) "abcdefg".match(/(?<=b).\*?(?=e)/)[0]  
→ "cd"
- reward関数内で取得した値は、正しく整形する必要があります。ScrapingWorkモジュールのreward関数を参考にしましょう。

## 【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

### ■ XPath取得方法

ブラウザ：Chrome

PC：Mac

対象のページの適当な部分で、右クリックを行うと、下記（左）のようにポップアップが出現します。ポップアップの「検証」をクリックすると、下記（右）のエリアが出現します。



次に、出現したエリアの左上のアイコンをクリックした状態で、XPathを取得したい要素付近をクリックします。

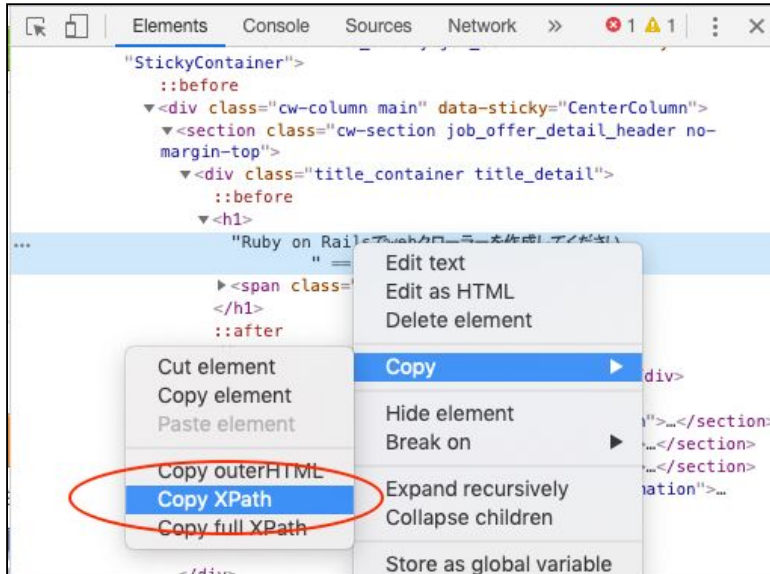


## 【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

すると、該当部分のHTMLが表示されますので、実際に取得する部分をクリックして、クリックした行の上で右クリックをすると、下記画像のポップアップが出現します。

「Copy」 → [Copy XPath]を選択するとクリップボードに該当要素のXPathがコピーされます。



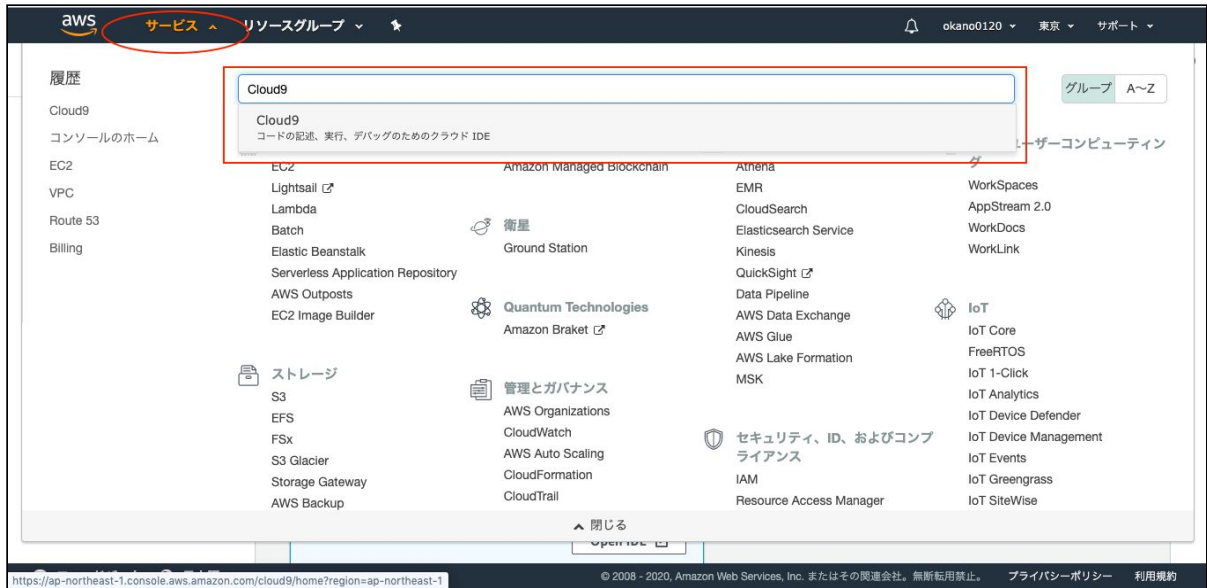
## 【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

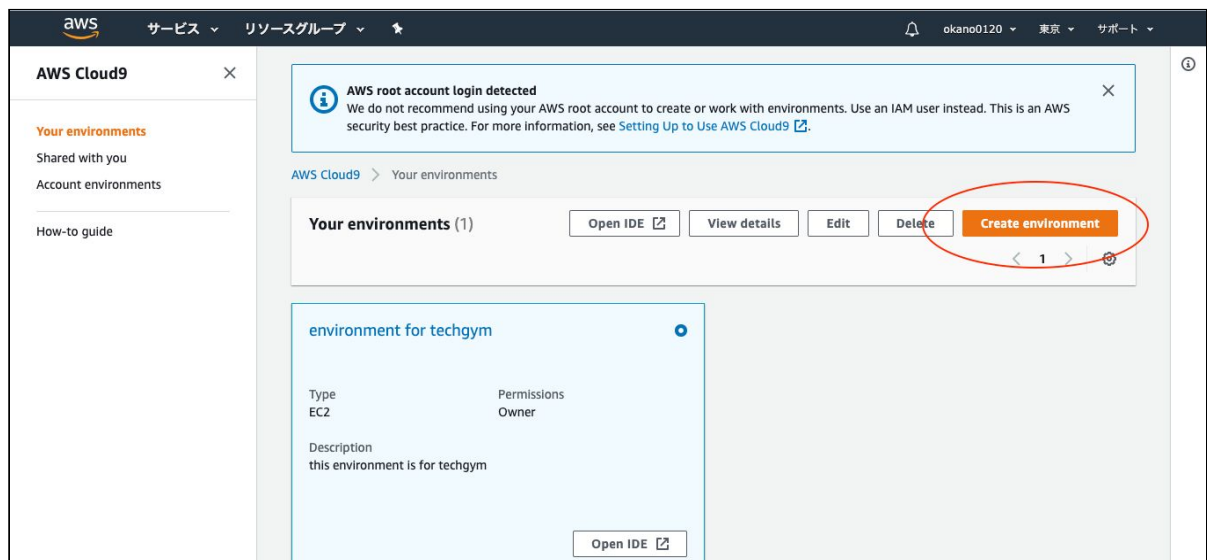
### ■ Cloud9の立ち上げ方

#### 【手順】

- ・ AWS(<https://aws.amazon.com/jp/>)にログインして、フッターの「サービス」をクリックし、検索フォームにCloud9と入力してます。すると、「Cloud9」の項目が出てくるので、クリックしてください。



- ・ Cloud9のダッシュボードに移動するので、「Create environment」をクリック



## 【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目 : / 分 2回目 : / 分 3回目 : / 分 4回目 : / 分 5回目 : / 分

- ・ Step 1 「Name environment」 では、好きな名前を入力し、任意で説明を入力してます。

AWS Cloud9 > Environments > Create environment

Step 1  
Name environment

Step 2  
Configure settings

Step 3  
Review

### Name environment

#### Environment name and description

**Name**  
The name needs to be unique per user. You can update it at any time in your environment settings.

techgym\_rails

Limit: 60 characters

**Description - Optional**  
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.

environment for techgym\_rails

Limit: 200 characters

- ・ Step 2 「Configure settings」 では、下記の内容を選択し、「Next step」をクリックして下さい。  
Environment type : Create a new instance for environment(EC2)  
Instance type : t2.micro(1 GiB RAM + 1 vCPU)  
Platform : Amazon Linux  
Cost-saving setting : After 30 minutes (default)

## Configure settings

### Environment settings

**Environment type** [Info](#)  
Choose between creating a new EC2 instance for your new environment or connecting directly to your server over SSH.

☒ **Create a new instance for environment (EC2)**  
Launch a new instance in this region to run your new environment.

☐ **Connect and run in remote server (SSH)**  
Display instructions to connect remotely over SSH and run your new environment.

**Instance type**

☒ **t2.micro (1 GiB RAM + 1 vCPU)**  
Free-tier eligible. Ideal for educational users and exploration.

☐ **t3.small (2 GiB RAM + 2 vCPU)**  
Recommended for small-sized web projects.

☐ **m5.large (8 GiB RAM + 2 vCPU)**  
Recommended for production and general-purpose development.

☐ **Other instance type**  
Select an instance type.

t3.nano

【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目 : / 分 2回目 : / 分 3回目 : / 分 4回目 : / 分 5回目 : / 分

Platform

☒ Amazon Linux

☐ Ubuntu Server 18.04 LTS

Cost-saving setting

Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default) ▼

IAM role

AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

► Network settings (advanced)

No tags associated with the resource.

Add new tag

You can add 50 more tags.

Cancel Previous step Next step

- ・ Step 3 「Review」 では、内容を確認し「Create environment」をクリックして下さい。

## 【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

### ■ Cloud9でブラウザを立ち上げる

- ・ ページ上部の「Preview」をクリックし、「Preview Running Application」をクリック。



- ・ Cloud9の画面上で、仮想的なブラウザが表示されますので、ブラウザ上部のBrowserの右隣にあるボタンをクリックしてください。すると、新規ブラウザが表示され、bundle exec rails serverで立ち上げたページを表示することができます。



### ■ Oops VFS connection does not exist と表示された場合

ブラウザが問題を起こしている可能性が高いので、ブラウザを変えていただく(講師はChromeを使用しています)か、シークレットモードで再度AWS・Cloud9にログインしていただけますと、エラーがなくなると思います。

## 【テックジム】 Railsコース 第2章 「ウェブサイトから情報を取得しよう」

1回目： / 分 2回目： / 分 3回目： / 分 4回目： / 分 5回目： / 分

### ■ gitについて

#### 【前提知識】

- ・ 修正： gitではファイルを修正すると、自動で修正部分・新規追加ファイルを認識します。
- ・ コミット： いくつかの修正をひとまとまりにしたものです。
- ・ ブランチ： コミットを順番にまとめたものです。

#### 【コマンド】

ブランチの一覧を表示する。

```
$ git branch
```

特定のブランチ(lesson1)に切り替える

```
$ git checkout lesson1
```

修正・新規ファイルの一覧を表示する。

```
$ git status
```

特定のファイル(app/controllers/contacts\_controller.rb)をコミットできる状態にする。

```
$ git add app/controllers/contacts_controller.rb
```

カレントディレクトリ内の全てのファイルをコミットできる状態にする。

```
$ git add .
```

コミットできる状態にした修正・新規ファイルを名前(フォーム送信機能 追加)をつけてコミットする

```
$ git commit -m "フォーム送信機能 追加"
```

コミットを順番に表示する。

```
$ git log
```

特定のファイル(app/controllers/contacts\_controller.rb)を修正する前の状態に戻す

```
$ git checkout app/controllers/contacts_controller.rb
```