

## 【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下さると助かります。

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 lesson9 remotes/origin/lesson9

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、16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 lesson8
```

```
$ git branch -D lesson9
```

```
$ git checkout -b lesson9 remotes/origin/lesson9
```

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

## ■ 2-9 : パスの一覧を取得しよう : lesson9

【問題】

ランサーズで「Rails」と検索し、表示された案件のパス一覧を取得しましょう。

取得するページの例)

<https://www.lancers.jp/work/search?keyword=Rails&page=3>

【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： 案件一覧ページのHTMLを取得し、Nokogiriオブジェクトに変換しましょう。

URL一覧を取得し文字列の配列を表示する

※ サイトの負荷を軽減するために、配列の要素数は10にしてください。

【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。"

["/work/detail/2926701", "/work/detail/2978009", "/work/detail/2947732", "/work/detail/2812214", " . . .

"/work/detail/1436202"]

true

"依頼のジャンル\nIT・通信・インターネット依頼の目的・背景\n\n\n"

Google 及び Yahooで特定キーワードを検索した際に表示され・・・"

## ■概要

【ヒント】

- HTML取得・Nokogiriオブジェクト変換は、ScrapingWorkLancersモジュールのget\_work\_docメソッドを利用する

- パスはaタグのhref属性より取得することができます。

- 部分配列を取得するためには、sliceメソッドを使用します。sliceメソッドは引数に範囲を取り、与えられた範囲の部分配列を返します。

例)

```
["a", "b", "c", "d", "e"].slice(0...3)
```

→ ["a", "b", "c"]

【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 - 10 : 複数の案件を保存しよう : lesson10

【はじめに】

```
$ git add .
$ git commit -m "パス一覧取得"
$ git checkout -b lesson10 remotes/origin/lesson10
```

【問題】

lesson9で取得した案件のパス一覧を元に、全ての案件を保存しましょう。  
ランサーズ以外のURLも表示される可能性があるので、ランサーズの案件のみ取得してください。

また、案件取得時に予測不可能なエラーが発生する可能性があり、エラーが発生した際に処理が止まらないように、Workモデルのcreate!メソッドをcreateメソッドに変更してください。

※ プログラムから特定のサイトに複数回アクセスする場合には、一回ごとに必ず5秒以上sleepさせてください。(サイトの負荷にならないようにするため)

【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 : 複数の案件情報を取得しデータベースに保存する。

【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	
21	ランサーズ	Ruby on Rails を使っ...	5000	6000	終了	<div>表示編集削除</div>
22	ランサーズ	AWS+Rubyを用いたシステムの...	20000	30000	終了	<div>表示編集削除</div>
23	ランサーズ	rails開発中アプリのdevis...	10000	20000	終了	<div>表示編集削除</div>
24	ランサーズ	Webアプリケーション開発における...	10000	20000	終了	<div>表示編集削除</div>
25	ランサーズ	動画学習サービスのAPI開発の仕事	300000	400000	終了	<div>表示編集削除</div>
26	ランサーズ	不動産CRM   顧客   物件   帳票出力...	300000	400000	終了	<div>表示編集削除</div>
27	ランサーズ	灯油宅配の会員制サイトの仕事	100000	200000	終了	<div>表示編集削除</div>
28	ランサーズ	webメール送信ツールの制作の仕事	100000	200000	終了	<div>表示編集削除</div>

【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】

- 5秒間sleepさせたい時には、下記の関数を実行する。

```
sleep(5)
```

- 配列の要素ごとに繰り返し処理を行いたい場合には、eachメソッドを利用します。

例) arrayは文字列の配列、elementにはarrayの各要素が順番に格納される。

```
array = ["a", "b", "c"]
```

```
array.each do |element|
```

```
  p element
```

```
end
```

```
→ "a"
```

```
"b"
```

```
"c"
```

- ランサーズのURLであるかどうかは、/work/detail/{数字}の形になっているかどうかで判別しましょう。

例)/work/detail/2926701

【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 - 11 : 案件情報を更新しましょう : lesson11

【はじめに】

```
$ git add .
```

```
$ git commit -m "複数案件保存機能追加"
```

```
$ git checkout -b lesson11 remotes/origin/lesson11
```

【問題】

取得した案件の情報がすでに保存されていた場合には、情報が更新されるようにしましょう。

【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 : すでに保存されているかどうかに応じて保存または更新する

【実行結果】

案件管理画面にアクセス(URL: /)

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

正しく更新できているかどうかを確認するために一度、データを変更する。

適当な案件の「編集」をクリックし、URL以外の内容を変更し「送信」をクリック。

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/>

データを変更した案件の「表示」をクリックし、変更した内容・更新日・作成日を確認。

①運用は、社内のみでできれば充分なのです。一般に公開するものではありません。弊社の共有サーバ内（PHP5.1.6、mySQL5が動く環境）のみで運用します。

②納品物は、プログラムファイル一式と、DBダンプファイルをお願いします。

以上です。

シンプルさ重視です。よろしくお願い致します。

動作確認についてですが、  
納品をいただきましたら、弊社内にて設定を行い動作確認した上で、修正点があれば連絡させていただくという形をお願いできればと思います。

修正は2回まで、などの条件のうえでの見積り歓迎です！

すみません、PHPバージョンを訂正させていただきます。5.3まで大丈夫です！  
よろしくお願い致します。

期限日	
ステータス	終了
作成日	2020/05/25 14:49
更新日	2020/05/25 17:21

編集戻る



【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】

□ Workモデルのfind\_byメソッドは引数にハッシュを取り、一致したデータが存在する場合には一致したデータ(1件)を返し、一致するデータが存在しない場合には、nilを返します。

例)

Work.find\_by(title: "Ruby on Railsでwebクローラーを作成してください")

→ Workオブジェクトまたは、nil

□ URLにより同じ案件かどうかを判別します。

□ Workモデルのfind\_byメソッドの戻り値を適当な変数に格納し、その値を元に更新するか・新規作成するかを判別します。

□ Workオブジェクトのupdateメソッドは引数にハッシュを取り、引数に与えられたハッシュを元にデータベースを更新します。

【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 - 12 : モジュールのクラス化 : lesson12

【はじめに】

```
$ git add .  
$ git commit -m "案件更新機能追加"  
$ git checkout -b lesson12 remotes/origin/lesson12
```

【問題】

2つモジュール(ScrapingWork・ScrapingWorkLancers)をクラスに変更し、同じ処理が複数箇所で定義されないようにしましょう。

【修正する内容】

ファイル : lib/scraping\_work.rb  
修正するタスク : fetch\_lancers  
修正する内容 : モジュールからクラスに変更

ファイル : lib/scraping\_work\_lancers.rb  
修正するタスク : fetch\_lancers  
修正する内容 : モジュールからクラスに変更  
不要な関数定義を削除

【ヒント】

□ クラスはclass ClassName ~ endのように定義します。

例)

```
class ScrapingWork  
end
```

□ 不要な関数はScrapingWorkLancersクラスのget\_work\_docメソッドとfetch\_workメソッドの2つ。

□ クラスは継承をすることで、基本クラスのメソッドを派生クラスで使うことができる。

□ クラスの継承は下記のように記述する

```
class 基本クラス < 派生クラス  
end
```

## 【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。

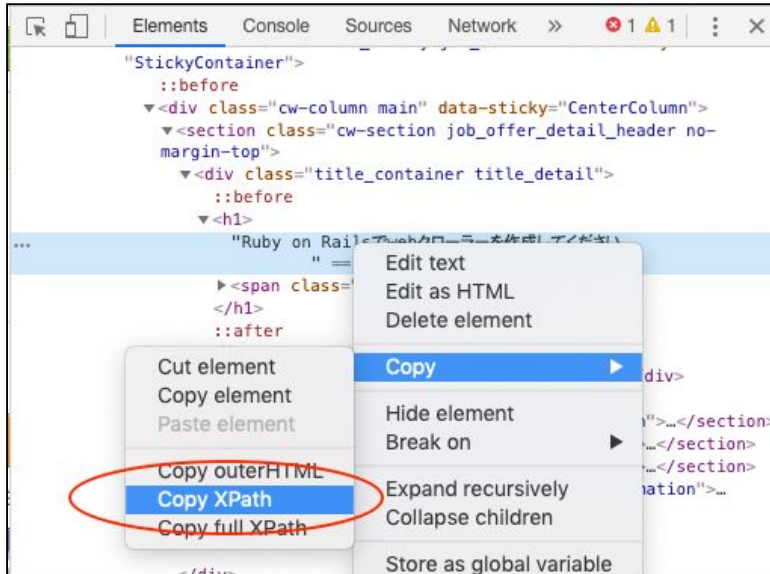


## 【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」

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。



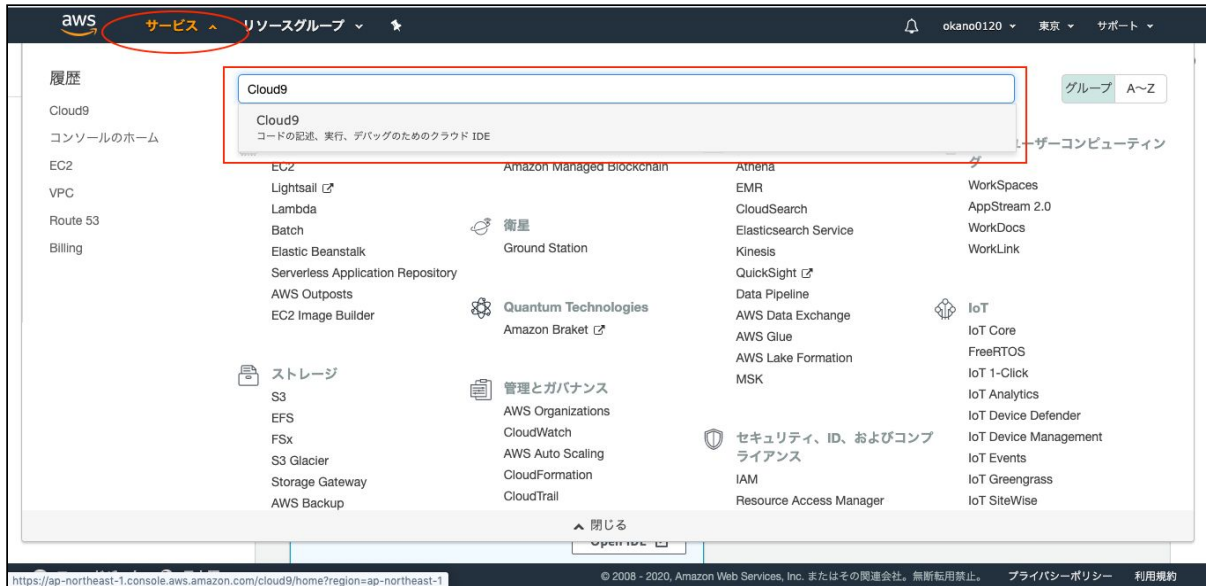
## 【テックジム】Railsコース 第2章 「ウェブサイトから情報を取得しよう」

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

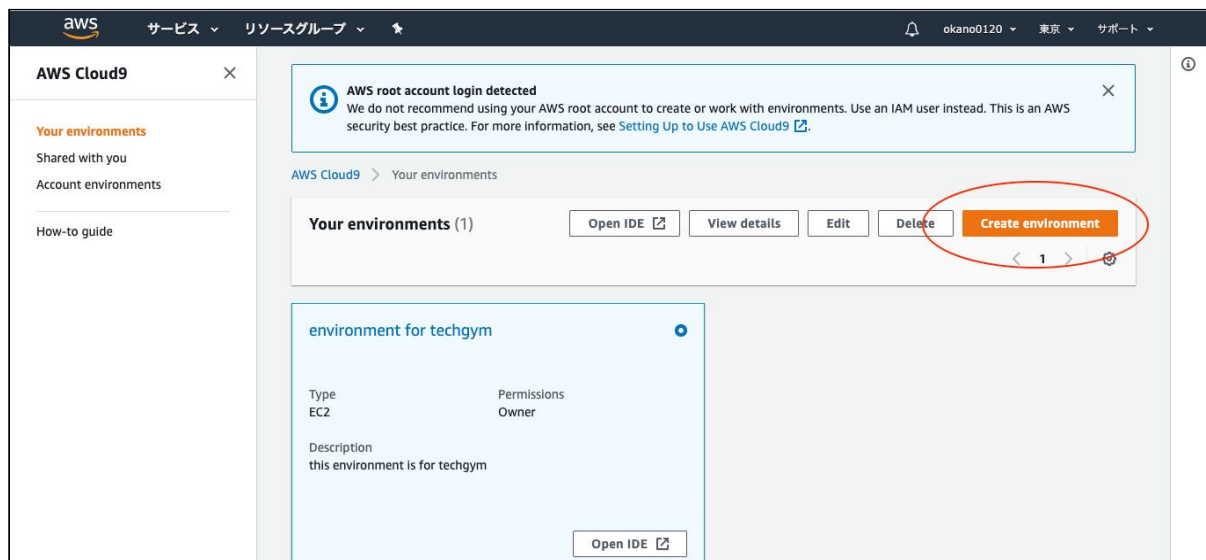
### ■ Cloud9の立ち上げ方

#### 【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**

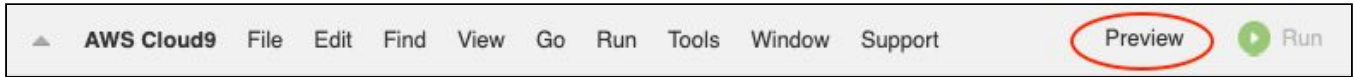
- ・ Step 3 「Review」 では、内容を確認し「Create environment」をクリックして下さい。

【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
```