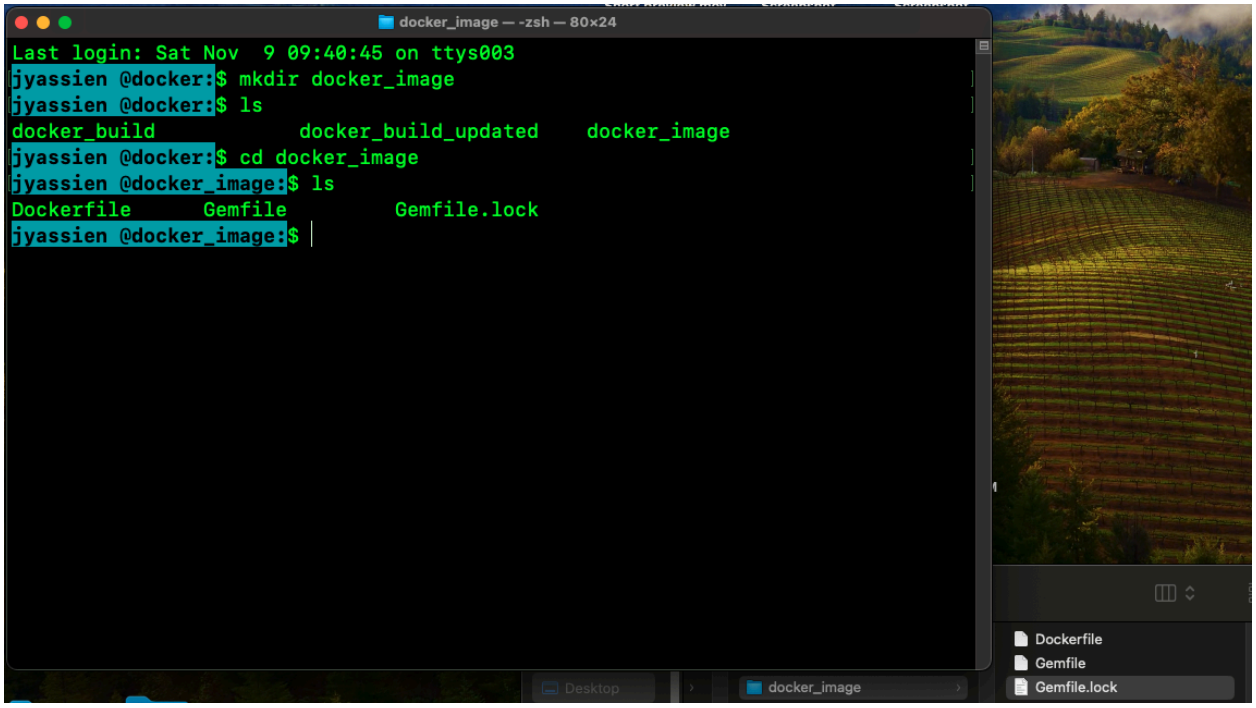
This documentation is about setting up a new Ruby on Rails project using Docker image from scratch:

Project name **Document**
Example models: **User and UserDocument**

Step 1: Create Docker Image:

- Create a folder and add docker image files
    - Dockerfile
    - Gemfile
    - Gemfile.lock



- Make sure you Docker Desktop is open before running the following image
    - Create an imge:
        - *docker build -t <DOCKER_USERNAME>/<NEW_IMAGE_NAME> .*
        - For my case:

```
docker build -t jyassiendock/ruby_rails_document_image .
```

```
jyassien @docker_image:$ docker build -t jyassiendock/ruby_rails_document_image .

[+] Building 2.1s (15/15) FINISHED                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile                                0.0s
 => => transferring dockerfile: 1.49kB                                              0.0s
 => resolve image config for docker-image://docker.io/docker/dockerfile:1           1.2s
 => [auth] docker/dockerfile:pull token for registry-1.docker.io                    0.0s
 => CACHED docker-image://docker.io/docker/dockerfile:1@sha256:865e5dd094beca432e8c0a1d5  0.0s
 => [internal] load metadata for registry.docker.com/library/ruby:3.2.5-slim        0.7s
 => [internal] load .dockerignore                                                   0.0s
 => => transferring context: 2B                                                     0.0s
 => [1/7] FROM registry.docker.com/library/ruby:3.2.5-slim@sha256:b304ff46af4dbf577c944d  0.0s
 => [internal] load build context                                                   0.0s
 => => transferring context: 3.25kB                                                 0.0s
 => CACHED [2/7] WORKDIR /workspace                                                 0.0s
 => CACHED [3/7] RUN curl -sL https://deb.nodesource.com/setup_16.x | bash - &&     apt-  0.0s
 => CACHED [4/7] RUN gem install bundler                                            0.0s
 => CACHED [5/7] COPY Gemfile Gemfile.lock ./                                       0.0s
 => CACHED [6/7] COPY . /workspace                                                  0.0s
 => CACHED [7/7] RUN bundle install                                                 0.0s
 => exporting to image                                                              0.0s
 => => exporting layers                                                             0.0s
 => => writing image sha256:cb1190e69b1945b0b85ce374ee1fa64c5b9217f98173622474f20ae0ec23  0.0s
 => => naming to docker.io/jyassiendock/ruby_rails_document_image                   0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/yvfeiwavym4bokmjpz44oke0r

 1 warning found (use docker --debug to expand):
 - FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 5)

What's next:
```

- Now let's run our newly created image. (You can run docker image anywhere in your directory)
- Let's create a new folder inside our docker_image folder called docker_container
  - `mkdir docker_container`
- Run docker image

```
jyassien @docker_container:$ docker run -it -p 3000:3000 -v $(pwd):/workspace jyassiendock/ruby_rails_document_image
root@2f74d07887fa:/workspace#
```

- The image has been initialized and a container was created with an ID 2f74d07887fa

Step 2: Create a Rails app:

- Let's create the **document** app with no bundle install and git initialization:
  - `rails new document --skip-bundle --skip-git`

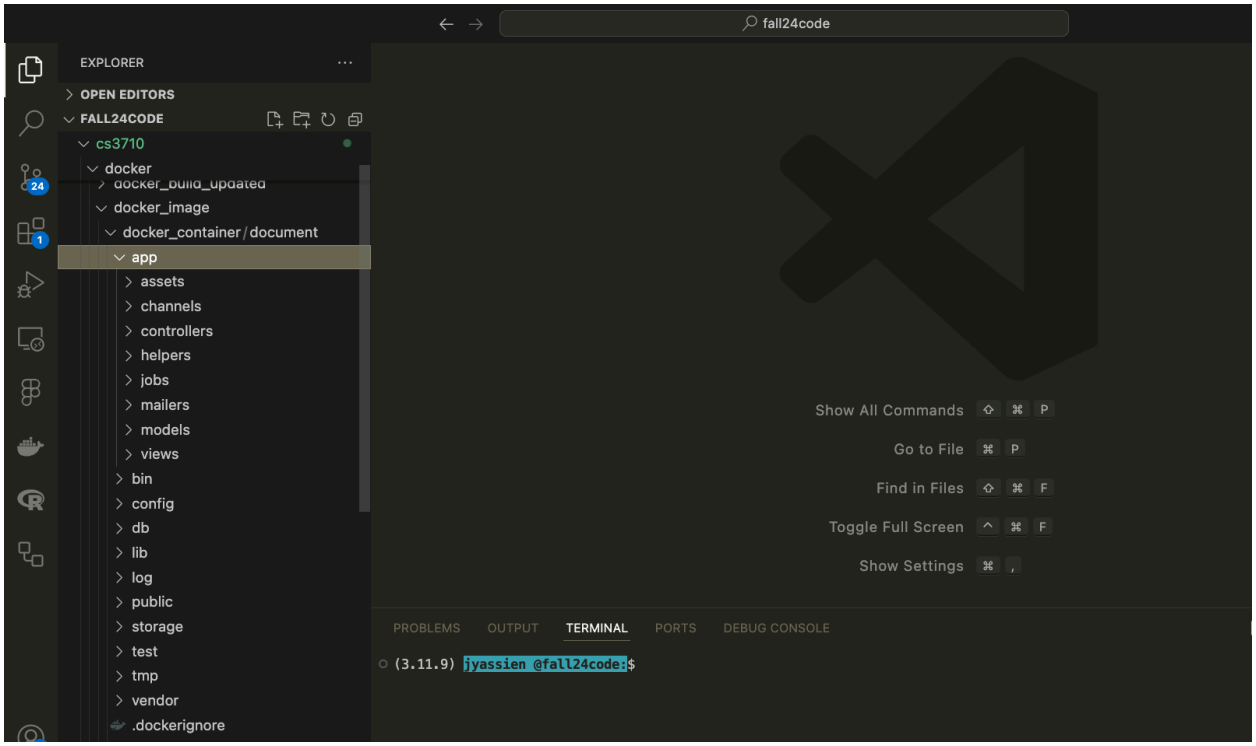
```
root@2f74d07887fa:/workspace# rails new document --skip-bundle --skip-git
Based on the specified options, the following options will also be activated:

  --skip-decrypted-diffs [due to --skip-git]

      create
      create  README.md
      create  Rakefile
      create  .ruby-version
      create  config.ru
      create  Gemfile
      create  app
      create  app/assets/config/manifest.js
      create  app/assets/stylesheets/application.css
      create  app/channels/application_cable/channel.rb
      create  app/channels/application_cable/connection.rb
      create  app/controllers/application_controller.rb
      create  app/helpers/application_helper.rb
      create  app/jobs/application_job.rb
      create  app/mailers/application_mailer.rb
      create  app/models/application_record.rb
      create  app/views/layouts/application.html.erb
      create  app/views/layouts/mailer.html.erb
      create  app/views/layouts/mailer.text.erb
      create  app/assets/images
      create  app/assets/images/.keep
      create  app/controllers/concerns/.keep
      create  app/models/concerns/.keep
      create  bin
      create  bin/rails
      create  bin/rake
```

- The document app is initialzaed with necessary folder structure

```
root@2f74d07887fa:/workspace# ls
document
root@2f74d07887fa:/workspace# cd document/
root@2f74d07887fa:/workspace/document# ls
Dockerfile  Gemfile  README.md  Rakefile  app  bin  config  config.ru  db  lib  log  public  storage  test  tmp  vendor
root@2f74d07887fa:/workspace/document#
```



Step 3: Set up two models with one-to-many relations;

A. Scaffold user model:-

```
rails generate scaffold User username:string email:string
subject:string
```

- *Since Docuemnt is a reserved word, I will user UserDocument.*

B. Scaffold UserDocument model with one-to-many relation ship with the user model:

```
rails generate scaffold UserDocument title:string content:text
user:references
```

C. Next step is migrating the database schemas

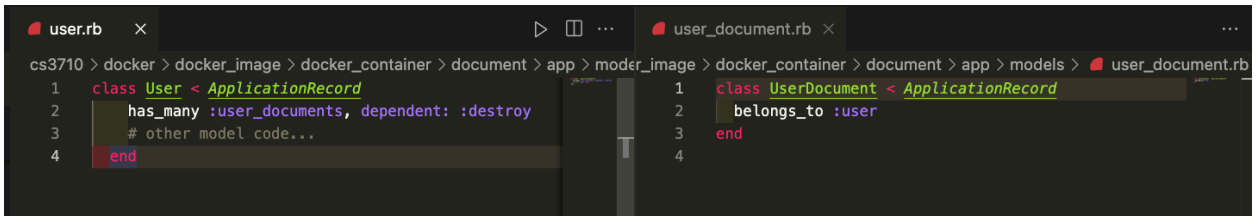
```
rails db:migrate
```

```
root@2f74d07887fa:/workspace/document# rails db:migrate
== 20241114032321 CreateUsers: migrating ======================================
-- create_table(:users)
   -> 0.0018s
== 20241114032321 CreateUsers: migrated (0.0018s) =============================

== 20241114032844 CreateUserDocuments: migrating ==============================
-- create_table(:user_documents)
   -> 0.0022s
== 20241114032844 CreateUserDocuments: migrated (0.0022s) =====================
```

D. Define model association:

```
class User < ApplicationRecord
  has_many :documents, dependent: :destroy
End
```



We have successful setted-up a Ruby on Rails app with one-to-many relationship from scratch using Docker Image.