**Requirements**

- Using Git and Heroku, deploy a Ruby on Rails application to the cloud.
- Secure communication between your app and users by enforcing SSL when users register and authenticate.

**Steps**

1. Create your own Heroku account by signing up at [https://api.heroku.com/signup](https://api.heroku.com/signup). Be sure to click the link in the confirmation email and create a password for your Heroku account.

2. Update your project’s [Gemfile](https://git-scm.com/docs/gemfile) to add support for Heroku. Find the line `gem 'sqlite3'` and replace it with the following.

   ```ruby
   group :development do
     gem 'heroku'
     gem 'sqlite3'
   end
   
   group :production do
     gem 'pg'
     gem 'rails_12factor'
   end
   ```

3. Tell Bundler to install all non-production gems by running the following command in Terminal.

   ```bash
   $ bundle --without production
   ```

4. Make the [heroku](https://github.com/joyent/heroku) command-line tool available by running the following command:

   ```bash
   $ bundle binstub heroku
   ```

5. You’ll be using Git to push your code to Heroku, but before you do, tell Git to ignore all installed gems. To do this, open the file `.gitignore` within your project’s root directory and add the following line to the end of the file. You may need to tell your editor to show hidden files before you’ll see the `.gitignore` file listed.

   ```bash
   /vendor/bundle
   ```

6. In Terminal, change your working directory to your project’s root directory and then run the following Git commands, in order. This will initialize a new Git repository for your project and commit all files in your project to the repository.

   ```bash
   $ git init
   $ git add .
   $ git commit -m "Initial commit"
   ```

7. Now you’re ready to start using Heroku. First, create a new Heroku app by running the following command in Terminal and entering your email and password when prompted. If prompted to generate a new public key, enter [y]. You should see output similar to the following. Notice the app name and URL that were created for you. If you open the URL in a browser, you should see “Welcome to your new app!”.

   ```bash
   $ bin/heroku create
   Enter your Heroku credentials.
   Email: <TYPE YOUR EMAIL>
   Password (typing will be hidden): <TYPE YOUR PASSWORD>
   Could not find an existing public key.
   Would you like to generate one? [Yn] y
   ```
Generating new SSH public key.
Uploading SSH public key /Users/djholt/.ssh/id_rsa.pub... done
Creating ancient-citadel-7378... done, stack is cedar
http://ancient-citadel-7378.herokuapp.com/ | git@heroku.com:ancient-citadel-7378.git
Git remote heroku added

8. [Optional] If you’d like to rename your app, run the following command and specify a new name for your app.

```bash
$ bin/heroku rename my-awesome-app
```

9. Now that you’ve created a Heroku app, deploy your Rails application to Heroku by running the following Git command. Read through the output and make sure there weren’t any errors or problems.

```bash
$ git push heroku master
```

10. Your Heroku app includes a brand new database that’s completely empty. Run the following command to apply all of your migrations to the new database.

```bash
$ bin/heroku run rake db:migrate
```

11. That’s it! Hit your app’s URL in a browser and see that everything works properly.

12. If you’d like to access the Rails console on your Heroku app (to set a user as an admin, for instance), run the following command.

```bash
$ bin/heroku run console
```

13. If you encounter errors or problems with your application after deploying to Heroku, run the following command to examine the logs for your application on Heroku. Since detailed error messages are no longer displayed in the web browser, you will need to refer to the application log for specific error messages.

```bash
$ bin/heroku logs
```

14. Now that your app is live and accessible to the public, it’s important to protect your users by securing any sensitive communication between your app and your users. So, tell your application to enforce SSL for all actions that handle user authentication and registration. To do this, simply add the following line to the top of `SessionsController` and `UsersController` (and any other controllers that handle sensitive information).

```ruby
force_ssl if Rails.env.production?
```

15. Now that you’ve made changes to your application, you’ll need to deploy the latest code to Heroku. Just like before, you’ll first need to commit your changes to your Git repository, and then push your changes to Heroku using Git. You can do this with the following commands.

```bash
$ git add .
$ git commit -m "Force SSL for user authentication and registration"
$ git push heroku master
```

**Submission**

- Show your completed assignment to the instructor during class or office hours to receive credit.