# Deploying Your App

| | |
|---|---|
| **Time** | ~ 2 - 3 hours |
| **Learning Goals** | • Learn how to read documentation.<br>• Learn how to push your ready app into an SasS platform. |

Congratulations for completing your very first web app! What you have created is known as *software-as-a-service (SaaS), which is a type of cloud service. It is time to show off your masterpiece to the world! To do that, we will need a platform to host your app.

There are many platform providers out in the internet, namely Heroku, Bluemix, OpenShift, Azure and so on. In this tutorial, we'll be using Heroku as our service provider is it is free and easy to use for Ruby and Rails developers.

As providers update their services consistently, this tutorial is created to guide you on how to source a particular provider's information, add the provider's configurations into your app settings, and launch your app into production.

This also means it requires A LOT of documentation reading. Also, please note that you should be using a Linux/MAC system to proceed.

## Sourcing Configurations From Provider

Heroku publishes their documentation here: https://devcenter.heroku.com/. We'll start our journey here. If you read through the tutorial, we'll be interested with "#deploy-the-app", "#define-a-Procfile", "#push-local-changes" and "#use-a-database". We'll go through each of them, step-by-step.

You'll also notice that Heroku uses Git to version control your software. This also means that we'll be using the terminal.

## Sign Up

For the first part, you should sign up for Heroku to gain access to Heroku services.

## Installing Toolbelt

Upon finding the keyword 'Ruby' and 'Rails' in the Heroku documentation page, you'll be guided into a training page with this URL address:

https://devcenter.heroku.com/articles/getting-started-with-ruby#set-up

This page asked us to install the toolbelt. In that case, go for it. You should be able to perform Heroku commands, such as the famous 'login'.

```
$ heroku login
```

# Deploying Your App

```
Enter your Heroku credentials.
Email: josh@example.com
Password:
```

## Deploying the app

We will first create a Heroku app in our local app repo, as guided by the #deploy-the-app.

1) In your terminal, go to your app folder.

```
$ cd ~/<Your app folder>
```

Important NOTE: The steps beyond here require you to be in your app folder. Heroku commands operate similarly to Git.

2) Perform a Heroku app creation using the following command:

```
$ heroku create
```

3) Rename your app to your desired app name:

```
$ heroku apps:rename
```

## Define a Procfile

As you read through the #define-a-procfile documentation, Heroku recommends PUMA as a web server and requires a Procfile to first execute the app.

1) Include `PUMA` and `RAILS_12Factor` into your Gemfile. Ensure that these gems are inside the `:production tag`. The code should look something like this:

```
group :production do
  # Heroku
  # =================
  gem 'rails_12factor' # Heroku Gem Supports
  gem 'puma' # Using puma for
end
```

2) Create a Procfile in the root folder. It should contain PUMA initialization as guided in the documentation.

```
web: bundle exec puma -t 5:5 -p ${PORT:-3000} -e ${RACK_ENV:-development}
```

3) Add the changes and commit it locally using Git.

```
$ git add .
$ git commit -m "adding Heroku support into framework"
```

## Push to Heroku

Now, we're ready to push to Heroku. Based on the #push-local-changes documentation, our last command would be git push to Heroku master branch.

1) Check the status of the app. Ensure it is clean and up to date.

```
$ git status
```

2) Push to Heroku master.

```
git push heroku master
```

3) Check for error messages. You can use the following command to query the status logs.

```
heroku logs
```

## Use a Database

Upon uploading the app successfully, you'll need to setup the database. By default, your framework provided by the Academy or Rails framework usually has the setup done for you. Hence, you don't need to perform "db:create" during production.

# Deploying Your App

1) Check your app status from the Heroku dashboard. Did it provide a database for your app? Usually, it does. Otherwise, read the documentation on how to proceed with “add-ons”.

2) Perform “rake db:migrate” on Heroku.

```
heroku run rake db:migrate
```

3) If you have seed files, perform seeding on Heroku.

```
heroku run rake db:seed
```

Now go to your website and check that everything is working as intended. If all is well, your app is ready for public use. Congratulations! You’re ready!

Discussions:

1. What is “heroku run”?
2. What can “heroku run” perform?
3. What database products do Heroku provide?
4. Did you actually read all the documents? What is the best way to extract information from these documentation centres?

Discuss among your peers and do some research on the Heroku documentation centre. This discussion will help you understand the available database technologies in the market and discover other ways of doing things for your future deployment.