

Introduction

Set up

Prepare the app

Deploy the app

View logs

Define a Procfile

Scale the app

Declare app dependencies

Run the app locally

Push local changes

Provision add-ons

Start a one-off dyno

Define config vars

Use a database

Next steps

Push local changes

In this step, you'll learn how to propagate a local change to the application through to Heroku. As an example, you'll modify the application to add an additional dependency and the code to use it.

Modify `Gemfile` to include a dependency for the `cowsay` gem. Your file will look something like this:

```
gem 'cowsay'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '4.1.4'
...
```

Modify `app/views/welcome/index.erb` so that it uses this gem. Your final code should look like this:

```
<h1>Getting Started with Ruby</h1>

<p>
  Welcome!
</p>

<pre>
<%= Cowsay.say("Hello") %>
</pre>
```

Now test locally:

```
> jruby -S bundle install
> heroku local web -f Procfile.windows
```

Visit your application at <http://localhost:5000> (<http://localhost:5000>). You should see an ASCII picture displayed.

Now, deploy this local change to Heroku.

Almost every deploy to Heroku follows the same pattern. First, add the modified files to your local Git repository:

```
> git add .
```

Now commit the changes to the repository:

```
> git commit -m "Demo"
```

Now deploy, just as you did previously:

```
> git push heroku master
```

Finally, check that everything is working:

```
> heroku open
```

[Log in to report a problem \(/login?back_to=%2Farticles%2Fgetting-started-with-jruby%23push-local-changes\)](/login?back_to=%2Farticles%2Fgetting-started-with-jruby%23push-local-changes)

I can push local changes (push-local-changes)

(Log in (/login?back_to=%2Farticles%2Fgetting-started-with-jruby%23push-local-changes) to save and track your progress)