



## **Chapter 2**

### **Lab 2.2**

**Objective: Install Ruby and create a “Hello World” application**

1. Prepare your environment by installing the pre-requisites:

```
$ sudo apt-add-repository ppa:brightbox/ruby-ng
$ sudo apt-get update && sudo apt-get upgrade
```

Next, install the Ruby Version Manager (RVM) to ensure that you are using the same version of Ruby, even if another version has already been installed:

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys
409B6B1796C275462A1703113804BB82D39DC0E3
$ curl -sSL https://get.rvm.io | bash -s stable
```

Source the RVM script for future commands:

```
$ source ~/.rvm/scripts/rvm
```

Use the following command to install a specific version of Ruby:

```
$ rvm install 2.3
```

2. Create the files needed for a simple Ruby application.

Make sure that the current working directory is the `hello` repository from Lab 2.1. `vi` and `Nano` are both installed and will work for this exercise. Create a file named `app.rb` and add the following lines:

```
$ require 'sinatra'

$ class HelloWorld < Sinatra::Base
$ get '/' do
  "Hello, world!"
end
$ get '/:name' do
  "Hello, #{params[:name]}!"
end
```

```
end  
end
```

Then, add another file named `config.ru`:

```
$ require './app'  
$ run HelloWorld
```

3. Dependencies will be managed with **bundler**.

Install **bundler** in the `hello` directory created in Lab 2.1:

```
$ gem install bundler
```

**Bundler** reads dependencies from a **Gemfile**. Using a preferred text editor, create a file named **Gemfile** in the `hello` directory, and add the following lines:

```
$ source 'https://rubygems.org'  
$ gem 'sinatra'  
$ gem 'minitest'  
$ gem 'rack-test'
```

Install from the **Gemfile** with:

```
$ bundle install
```

4. The web application can be tested with the following **rackup** command:

```
$ rackup -o 0.0.0.0
```

Using a web browser, navigate to `http://<your ip>:9292`. The “Hello World” application should be displayed in the browser.

You can also use `http://<your ip>:9292/something`.

“Hello, something” should be the output to the page. The application takes whatever is in the first directory in the path component and appends that to the “Hello” statement. You will use this for testing.

**Note:** If you are using Vagrant with VirtualBox, the IP address will serve as the primary interface of your host machine. For example, on a MAC running Vagrant, it should be the address for the `en0` interface from an `ifconfig -a` command.

5. Another file, `test.rb`, will be added with the following contents:

```
$ require './app.rb'
$ require 'minitest/autorun'
$ require 'rack/test'

$ class MyAppTest < Minitest::Test
  include Rack::Test::Methods

  def app
    HelloWorld
  end

  def test_my_default
    get '/'
    assert_equal 'Hello, world!', last_response.body
  end

  def test_with_params
    get '/Frank'
    assert_equal 'Hello, Frank!', last_response.body
  end
end
```

6. Use the following command to run this test:

```
$ ruby test.rb
```

Your output should look like this:

```
Run options: --seed 13471

# Running:

..
Finished in 0.015424s, 129.6701 runs/s, 129.6701 assertions/s.
2 runs, 2 assertions, 0 failures, 0 errors, 0 skips
```

7. Finally, commit the code, and push it up to your GitHub repository:

```
$ git add .
$ git commit -m "Initial app commit"
$ git push origin master
```