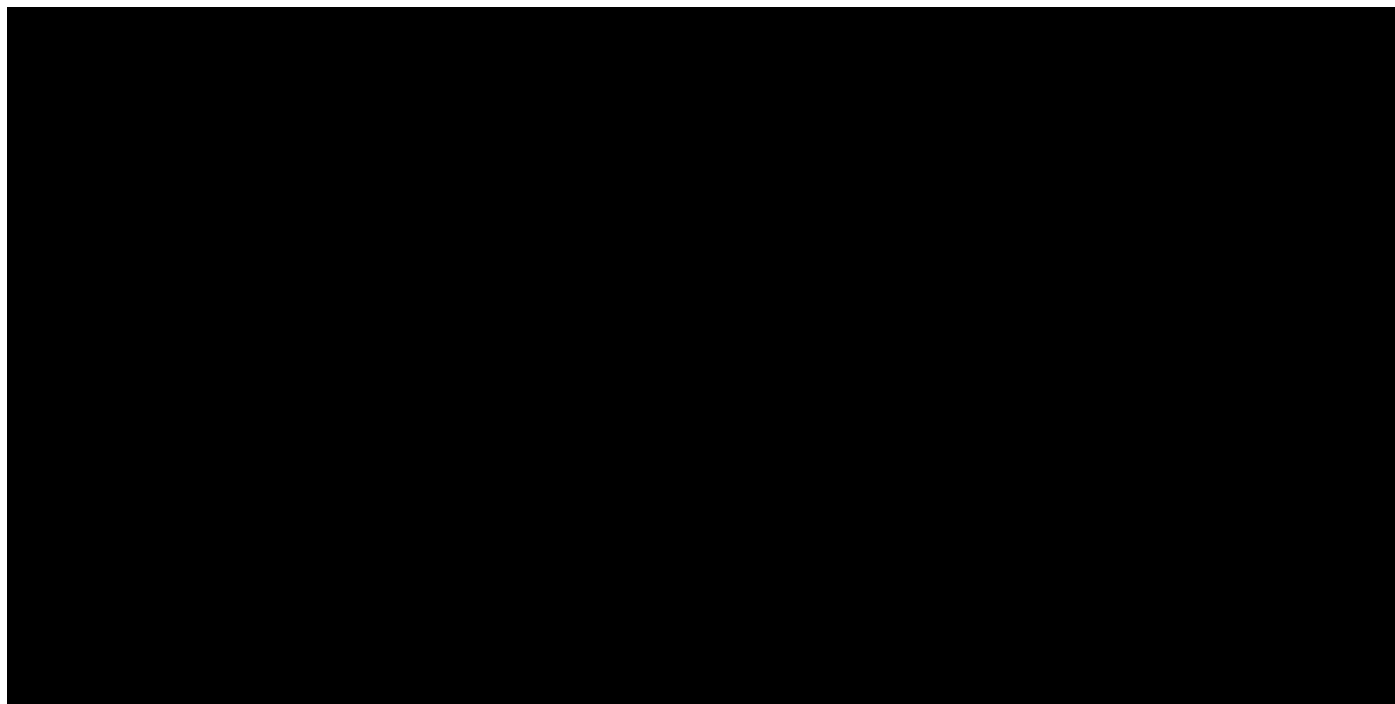


By: Mitchell Anicas

 Subscribe Share Contents ▼

How To Deploy a Rails App with Puma and Nginx on Ubuntu 14.04

Posted Apr 1, 2015  23  58.7k [Deployment](#) [Ruby on Rails](#) [Nginx](#) [PostgreSQL](#) [Ruby](#) [Ubuntu](#)

Introduction

When you are ready to deploy your Ruby on Rails application, there are many valid setups to consider. This tutorial will help you deploy the production environment of your Ruby on Rails application, with PostgreSQL as the database, using Puma and Nginx on Ubuntu 14.04.

Puma is an application server, like Passenger or Unicorn, that enables your Rails application to process requests concurrently. As Puma is not designed to be accessed by users directly, we will use Nginx as a reverse proxy that will buffer requests and responses between users

and your Rails application.

Prerequisites

This tutorial assumes that you have an Ubuntu 14.04 server with the following software installed, on the user that will deploy the application:

- [Ruby on Rails, using rbenv](#)
- [PostgreSQL with Rails](#)

If you do not have that set up already, follow the tutorials that are linked above. We will assume that your user is called **deploy**.

Also, this tutorial does not cover how to set up your development or test environments. If you need help with that, follow the example in the [PostgreSQL with Rails tutorial](#).

Create Rails Application

Ideally, you already have a Rails application that you want to deploy. If this is the case, you may skip this section, and make the appropriate substitutions while following along. If not, the first step is to create a new Rails application that uses PostgreSQL as its database.

This command will create a new Rails application, named "appname" that will use PostgreSQL as the database. Feel free to substitute the highlighted "appname" with something else:

```
rails new appname -d postgresql
```

Then change into the application directory:

```
cd appname
```

Let's take a moment to create the PostgreSQL user that will be used by the production

environment of your Rails application.

Create Production Database User

To keep things simple, let's name the production database user the same as your application name. For example, if your application is called "appname", you should create a PostgreSQL user like this:

```
sudo -u postgres createuser -s appname
```

We want to set the database user's password, so enter the PostgreSQL console like this:

```
sudo -u postgres psql
```

Then set the password for the database user, "appname" in the example, like this:

```
\password appname
```

Enter your desired password and confirm it.

Exit the PostgreSQL console with this command:

```
\q
```

Now we're ready to configure the your application with the proper database connection information.

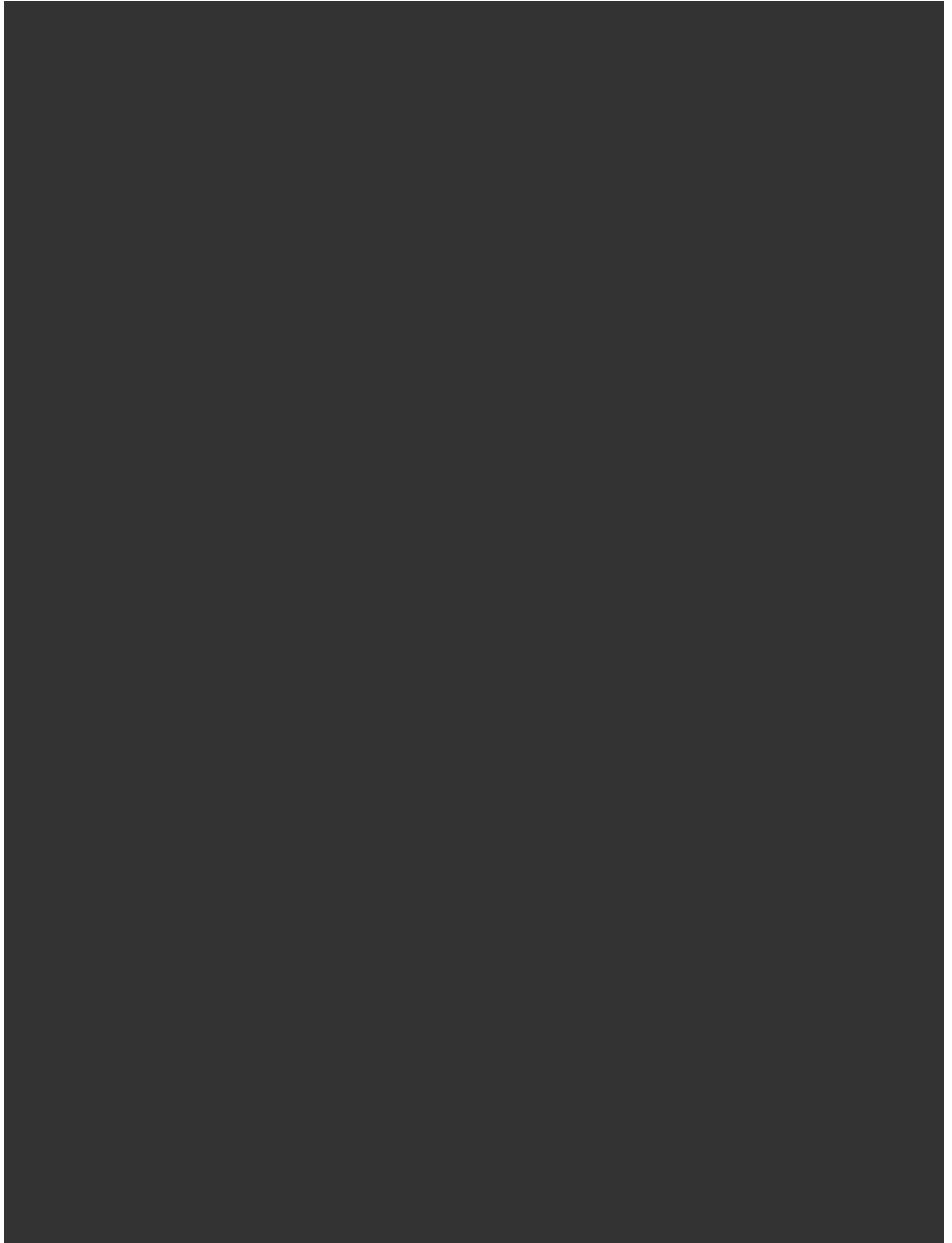
Configure Database Connection

Ensure that you are in your application's root directory (`cd ~/appname`).

Open your application's database configuration file in your favorite text editor. We'll use vi:

```
vi /etc/mysql/mysql.conf.d/mysqld.cnf
```

Up



database user password):

```
APPNAME_DATABASE_PASSWORD=prod_db_pass
```

Save and exit.

You may view which environment variables are set for your application with the `rbenv-vars` plugin by running this command:

```
rbenv vars
```

If you change your secret or database password, update your `.rbenv-vars` file. Be careful to keep this file private, and don't include it any public code repositories.

Create Production Database

Now that your application is configured to talk to your PostgreSQL database, let's create the production database:

```
RAILS_ENV=production rake db:create
```

Generate a Controller

If you are following along with the example, we will generate a scaffold controller so our application will have something to look at:

```
rails generate scaffold Task title:string note:text
```

Now run this command to update the production database:

```
RAILS_ENV=production rake db:migrate
```

You should also precompile the assets:

```
RAILS_ENV=production rake assets:precompile
```

To test out if your application works, you can run the production environment, and bind it to the public IP address of your server (substitute your server's public IP address):

```
RAILS_ENV=production rails server --binding=server_
```

At the end of the file, add the Puma gem with this line:

```
gem 'puma'
```

Save and exit.

To install Puma, and any outstanding dependencies, run Bundler:

```
bundle
```

Puma is now installed, but we need to configure it.

Configure Puma

Before configuring Puma, you should look up the number of CPU cores your server has. You can easily do that with this command:

```
grep -c processor /proc/cpuinfo
```

Now, let's add our Puma configuration to `config/puma.rb`. Open the file in a text editor:

```
vi config/puma.rb
```

Copy and paste this configuration into the file:

```
# Change to match your CPU core count  
workers 2
```

```
# Min and Max threads per worker  
threads 1, 6
```

```
app_dir = File.expand_path("../..", __FILE__)
```



```
shared_dir = "#{app_dir}/shared"

# Default to production
rails_env = ENV['RAILS_ENV'] || "production"
environment rails_env

# Set up socket location
bind "unix://#{shared_dir}/sockets/puma.sock"

# Logging
stdout_redirect "#{shared_dir}/log/puma.stdout.log", "#{shared_dir}/log/puma.stderr.log"

# Set master PID and state locations
pidfile "#{shared_dir}/pids/puma.pid"
state_path "#{shared_dir}/pids/puma.state"
activate_control_app

on_worker_boot do
  require "active_record"
  ActiveRecord::Base.connection.disconnect! rescue ActiveRecord::ConnectionNotEstablished
  ActiveRecord::Base.establish_connection(YAML.load_file("#{app_dir}/config/database.yml")
end
```

Change the number of `workers` to the number of CPU cores of your server.

Save and exit. This configures Puma with the location of your application, and the location of its socket, logs, and PIDs. Feel free to modify the file, or add any other options that you require.

Now create the directories that were referred to in the configuration file:

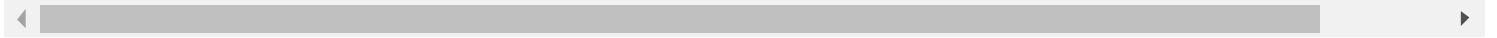
```
mkdir -p shared/pids shared/sockets shared/log
```

Create Puma Upstart Script

Let's create an Upstart init script so we can easily start and stop Puma, and ensure that it will

start on boot.

Downloû



Each line in this file should be the path to an application that you want `puma-manager` to manage. Add the path to your application now. For example:

```
/home/deploy/appname
```

Save and exit.

Now your application is configured to start at boot time, through Upstart. This means that your application will start even after your server is rebooted.

Start Puma Applications Manually

To start all of your managed Puma apps now, run this command:

```
sudo start puma-manager
```

You may also start a single Puma application by using the `puma` Upstart script, like this:

```
sudo start puma app=/home/deploy/appname
```

You may also use `stop` and `restart` to control the application, like so:

```
sudo stop puma-manager  
sudo restart puma-manager
```

Now your Rails application's production environment is running under Puma, and it's listening on the `shared/sockets/puma.sock` socket. Before your application will be accessible to an outside user, you must set up the Nginx reverse proxy.

Install and Configure Nginx

Install Nginx using `apt-get`:

```
sudo apt-get install nginx
```

Now open the default server block with a text editor:

```
sudo vi /etc/nginx/sites-available/default
```

Replace the contents of the file with the following code block. Be sure to replace the highlighted parts with the appropriate username and application name (two locations):

```
upstream app {  
    # Path to Puma SOCK file, as defined previously  
    server unix:/home/deploy/appname/shared/sockets/puma.sock fail_timeout=0;  
}
```

```
server {  
    listen 80;  
    server_name localhost;  
  
    root /home/deploy/appname/public;
```

Sign up for our newsletter.



Get the latest tutorials on SysAdmin and open source topics.

Sign Up

```
proxy_redirect off;  
}
```

SCROLL TO TOP

```
error_page 500 502 503 504 /500.html;  
client_max_body_size 4G;  
keepalive_timeout 10;  
}
```

Save and exit. This configures Nginx as a reverse proxy, so HTTP requests get forwarded to

the Puma application server via a Unix socket. Feel free to make any changes as you see fit.

Restart Nginx to put the changes into effect:

```
sudo service nginx restart
```

Now the production environment of your Rails application is accessible via your server's public IP address or FQDN. To access the Tasks controller that we created earlier, visit your application server in a web browser:

```
http://server_public_IP/tasks
```

You should see the same page that you saw the first time you tested your application, but now it's being served through Nginx and Puma.

Conclusion

Congratulations! You have deployed the production environment of your Ruby on Rails application using Nginx and Puma.

If you are looking to improve your production Rails application deployment, you should check out our tutorial series on [How To Use Capistrano to Automate Deployments](#). The series is based on CentOS, but it should still be helpful in automating your deployments.



Author:
Mitchell Anicas

Spin up an SSD cloud server in under a minute.

Simple setup. Full root access. Straightforward pricing.

DEPLOY SERVER

Related Tutorials

How To Deploy a Rails App with azk

Blueprint: How Ghost Migrated From Dedicated Servers to DigitalOcean

How To Deploy a Symfony Application to Production on Ubuntu 14.04

How To Deploy a Hugo Site to Production with Git Hooks on Ubuntu 14.04

How To Use Blue-Green Deployments to Release Software Safely

40 Comments

Leave a comment...

Logged in as:

Notify me of replies
to my comment

Comment

meghanhartman11 *April 3, 2015*

I'm getting this error: "Job failed to start" when I run sudo start puma-manager.

manicas *April 27, 2015*

Ensure that that the contents of `/etc/init/puma.conf` and `/etc/init/puma-manager.conf` are correct. This is covered in the [Create Puma Upstart Script section](#).

slavajacobson *August 1, 2015*

Getting the same error. I copied all 3 confs exactly as they are into the correct locations, and changed user and group to the correct one.

sudo start puma.conf app=/location/to/app works fine but sudo start puma-manager gives me the "Job failed to start" error.

Were you able to solve it?

luciusyg *April 26, 2015*

Hi I get the error "start: Job failed to start" when i typed sudo start puma-manager

manicas *April 27, 2015*

Ensure that that the contents of `/etc/init/puma.conf` and `/etc/init/puma-manager.conf` are correct. This is covered in the [Create Puma Upstart Script section](#).

sabernar *April 29, 2015*

My site loads, but when I click a link, I get the following error:

```
2015/04/29 21:47:20 [crit] 5512#0: *120 connect() to
unix:/var/www/myapp/shared/sockets/puma.sock failed (2: No such file or directory) while
connecting to upstream, client: 96.125.117.12, server: localhost, request: "GET /pages/faq.html
HTTP/1.1", upstream: "http://unix:/var/www/myapp/shared/sockets/puma.sock:/pages/faq.html",
host: "myserver.amazonaws.com", referer: "http://myserver.amazonaws.com/"
```


Yes bundler is installed

I am using rvm

```
rvm -v
```

```
rvm 1.26.11 (latest) by Wayne E. Seguin wayneeseguin@gmail.com, Michal Papis  
mpapis@gmail.com [https://rvm.io/]
```

```
deploy@akticiti:~/apps/akticiti.com$ ruby -v
```

```
ruby 2.2.1p85 (2015-02-26 revision 49769) [x86_64-linux]
```

```
deploy@akticiti:~/apps/akticiti.com$ rails -v
```

```
Rails 4.2.2
```



saibharath *August 4, 2015*

`http://server-ip/tasks` gives me a 403 Forbidden

Here is `ls -l` of my public folder

```
-rw-r--r-- 1 root root 1564 Aug 4 02:27 404.html  
-rw-r--r-- 1 root root 1547 Aug 4 02:27 422.html  
-rw-r--r-- 1 root root 1477 Aug 4 02:27 500.html  
drwxr-xr-x 2 root root 4096 Aug 4 03:01 assets  
-rw-r--r-- 1 root root 0 Aug 4 02:27 favicon.ico  
-rw-r--r-- 1 root root 202 Aug 4 02:27 robots.txt
```



manicas *August 4, 2015*

Make sure that puma is running (`ps aux | grep [p]uma` should return something if it is).

Also, `appdir/shared/sockets/puma.sock` should exist.

After you check those things, check your Nginx configuration and make sure all of the paths are correct. Restart Nginx if you change anything.



saibharath *August 4, 2015*

I do the following and still there is no puma.sock created.

```
sudo start puma app=apppath
```

```
puma (apppath) start/running, process 30059
```

```
ps aux|grep [p]uma
```

Returns nothing

Now the app throws,
502 Bad Gateway instead of 403

lsof -i gives the following output

```
nginx 31087 root 6u IPv4 219652 0t0 TCP *:http (LISTEN)
nginx 31089 www-data 6u IPv4 219652 0t0 TCP *:http (LISTEN)
nginx 31090 www-data 6u IPv4 219652 0t0 TCP *:http (LISTEN)
nginx 31091 www-data 6u IPv4 219652 0t0 TCP *:http (LISTEN)
nginx 31092 www-data 6u IPv4 219652 0t0 TCP *:http (LISTEN)
```

♡ 1

manicas *August 4, 2015*

Puma probably isn't configured correctly. You might want to check your app's logs (`app_path/shared/log/puma.stderr.log` and `puma.stdout.log`) for some clues.

♡

gonzalog *August 31, 2015*

I was in a similar situation. My problem was that I was supposed to edit '`/etc/nginx/conf.d/default.conf`' instead of '`/etc/nginx/sites-available/default`'.

♡

zykerboee *September 30, 2015*

did you find any solution? I got the same exact problem. Followed the instructions, If I execute **puma** in the rails project folder then it successfully runs, but application fails to load in the browser with "something wrong might have happened". PID, log files and socket are not created in folders. Nginx user is the same as the puma user and the same that has full permissions in the folder (vagrant box).

**connect() to
unix:///usr/share/nginx/html/rails_project/shared/sockets/puma.sock
failed (2: No such file or directory) while connecting to upstream**

♡

saibharath *September 30, 2015*

Just to make my sample app running I changed the user to root instead of www-data. There is some problem with the user www-data not being able to create those files. I am going to change that in future because the process shouldn't run as a root user. Try it out.

♡

saibharath *August 4, 2015*

The log file isn't created as well

cat: app_path/shared/log/puma.stderr.log: No such file or directory.

puma.rb in the config folder is configured with the same code given in this post

vi /etc/init/puma.conf

If I just run the last line then the server starts

exec bundle exec puma -C config/puma.rb

[32492] * Version 2.12.3 (ruby 2.2.1-p85), codename: Plutonian Photo Shoot

[32492] * Min threads: 1, max threads: 6

[32492] * Environment: production

[32492] * Process workers: 1

[32492] * Phased restart available

[32492] * Listening on unix:///app_path/shared/sockets/puma.sock

Even after that http://server_ip/tasks

throws 502



saibharath *August 4, 2015*

This is what I saw in syslog

Aug 4 16:32:38 hostname kernel: [80385.104593] init: puma (apppath) main process (2319) terminated with status 127

Aug 4 16:32:38 hostname kernel: [80385.104616] init: puma (apppath) main process ended, respawning

Aug 4 16:33:13 hostname kernel: [80419.713430] init: puma (app_path) respawning too fast, stopped



saibharath *August 4, 2015*

Aug 4 16:05:58 hostname puma-manager: Starting apppath

Aug 4 16:05:58 hostname puma: Starting server: apppath

Aug 4 16:05:59 hostname kernel: [78785.329427] init: puma (apppath) main process (1499) terminated with status 127

Aug 4 16:05:59 hostname kernel: [78785.329474] init: puma (apppath) main process ended, respawning

Aug 4 16:05:59 hostname puma: Starting server: /root/dragonball

Aug 4 16:05:59 hostname kernel: [78785.613801] init: puma (apppath) main process (1507)

terminated with status 127

Aug 4 16:05:59 hostname kernel: [78785.613815] init: puma (apppath) main process ended, respawning

Aug 4 16:05:59 hostname puma: Starting server: /root/dragonball

Aug 4 16:05:59 hostname kernel: [78785.824421] init: puma (apppath) main process (1514) terminated with status 127

Aug 4 16:05:59 hostname kernel: [78786.207507] init: puma (apppath) main process (1521) terminated with status 127

Aug 4 16:05:59 hostname kernel: [78786.207587] init: puma (app_path) respawning too fast, stopped



saibharath *August 4, 2015*

Modified the puma.conf and the server started finally by running puma command

I changed the following:

```
elif [ -d "$HOME/.rbenv/versions/2.2.1/bin" ]; then
```

```
export PATH="$HOME/.rbenv/versions/2.2.1/bin:$HOME/.rbenv/shims:$PATH"
```

Even after the server started

http://server_ip/tasks throws 502 error



saibharath *August 5, 2015*

This is the actual problem with nginx not able to connect to socket.

*2015/08/05 02:50:52 [crit] 20297#0: *1 connect() to unix:///apppath/shared/sockets/puma.sock failed (13: Permission denied) while connecting to upstream*

*2015/08/05 02:50:52 [crit] 20297#0: *1 stat() "/apppath/public/500.html/index.html" failed (13: Permission denied), client: 24.225.104.110, server: localhost*

request: "GET /favicon.ico HTTP/1.1", upstream:

"http://unix:///app_path/shared/sockets/puma.sock/favicon.ico", host: "45.55.246.50", referrer:

"http://45.55.246.50/500.html"

I checked permissions and gave 755 to all the files in public directory.

```
chmod -R 755 public/
```

Still the upstream says Permission denied

I did this next:

```
ps aux | grep "nginx: worker process"
```

```
www-data 20206 0.0 0.3 86160 1776 ? S 02:46 0:00 nginx: worker process
```

```
www-data 20207 0.0 0.4 86160 2276 ? S 02:46 0:00 nginx: worker process
www-data 20208 0.0 0.3 86160 1776 ? S 02:46 0:00 nginx: worker process
www-data 20209 0.0 0.3 86160 1776 ? S 02:46 0:00 nginx: worker process
```

So I changed the owner of the directory to www-data. Still doesn't work and same error as above



manicas *August 5, 2015*

I'm not sure exactly how you have it set up, but does the www-data user have the proper permissions to get to the app's directory?

Did you set the values of `setuid` and `setgid` in `/etc/init/puma.conf`?



waynesie92 *August 12, 2015*

Hey Mitchel. After typing this `sudo -u postgres createuser -s MyAppName`, I get `sudo: unknown user: postgres`
`sudo: unable to initialize policy plugin`. Do you have any clue to this error?



manicas *August 12, 2015*

Hi. You probably need to install the PostgreSQL package with `apt-get` (one of the prerequisite links). Run these commands:

```
$ sudo apt-get update
$ sudo apt-get install postgresql postgresql-contrib libpq-dev
```



jooseph *August 17, 2015*

Hello Mitchell I am a beginner and I am ready to deploy my Rails app that I developed on my local machine. So far on my droplet I set up all the required prerequisites. My question is, how do I transfer the app from my machine to the droplet? Should I upload the code to a repository like Bitbucket and and clone it on the droplet? Also would I have to create a new database on postgresql or does cloning the repo handle that for me. Thank you for writing these articles. I am learning a lot and they're the very reason I am using digital ocean!



manicas *August 17, 2015*

Hey there. Thanks for the kind words. I'm glad that you're enjoying the tutorials.

I think if you're deploying for the first time out, the way that you described (clone from a git repo) is the easiest way to copy your app to your server. For the database to be created, you need to create the PostgreSQL user (it's called "appname" in this tutorial), configure **database.yml** with the connection information, then run the **rake** tasks to create the database, then run the rake migrations.

After you understand that process, you may want to automate your deploy process with [Git Hooks](#) or [Capistrano](#).

♡ 1 romjjqThrdImomjjqTóæ7ljTksqsmTkTssTóå7ã7ljTksojjTkTlsmTóå7ã7hmnojsrjnrThmnojs

carlsonchrisw August 28, 2015r

Hi Mitchell,

Thanks for writing this tutorial. I've followed all the steps and am using AWS EC2 for my server.

When I try to go to the public ip I get this error.

This webpage is not available
ERRCONNECTIONREFUSED

I've been trying to [connect to the public ip](#) but I get this error. I've been trying to [connect to the public ip](#) but I get this error.

`http://127.0.0.1:8080` in a web browser.



smithcarlton *December 4, 2015*

Is there anything to indicate how to alter this if you need to use SSL?



Load More Comments



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



Copyright © 2016 DigitalOcean™ Inc.

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [RSS](#) 

[Distros & One-Click Apps](#) [Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Get Paid to Write](#)