How do you find a free TCP server port using Ruby?

Asked 16 years, 2 months ago Modified 3 years, 1 month ago Viewed 10k times

15

I'm trying to create a use-once HTTP server to handle a single callback and need help with finding a free TCP port in Ruby.

This is the skeleton of what I'm doing:

```
require 'socket'
t = STDIN.read
port = 8081
while s = TCPServer.new('127.0.0.1', port).accept
  puts s.gets
  s.print "HTTP/1.1 200/OK\rContent-type: text/plain\r\n\r\n" + t
  s.close
  exit
end
```

(It echoes standard input to the first connection and then dies.)

How can I automatically find a free port to listen on?

This seems to be the only way to start a job on a remote server which then calls back with a unique job ID. This job ID can then be queried for status info. Why the original designers couldn't just return the job ID when scheduling the job I'll never know. A single port cannot be used because conflicts with multiple callbacks may occur; in this way the ports are only used for +- 5 seconds.

ruby http sockets

Share
Improve this question
Follow

edited Oct 14, 2008 at 9:56

asked Oct 14, 2008 at 9:20
Marius Marais
532 ●2 ●6 ●12

6 Answers

Sorted by: Highest score (default)

69

Pass 0 in for the port number. This will cause the system to pick a port for you out of the ephemeral port range. Once you create the server, you can ask it for its addr, which will contain the port that the server is bound to.

```
server = TCPServer.new('127.0.0.1', 0)
port = server.addr[1]
```

Share Improve this answer Follow

answered Oct 14, 2008 at 15:00

Aaron Hinni
14.7k ● 6 ● 41 ● 39

I'm using a basic Socket (not TCPSocket), which doesn't have the `addr` method. Would you happen to know how I can get the port in this case? – troelskn Aug 19, 2009 at 16:28

3 Answering my own question, use `socket.getsockname.unpack("snA*")[1]` – troelskn Aug 19, 2009 at 17:02

3

It is actually quite easy when you don't try to do everything in one line :-/

```
require 'socket'
t = STDIN.read

port = 8080 # preferred port
begin
  server = TCPServer.new('127.0.0.1', port)
rescue Errno::EADDRINUSE
  port = rand(65000 - 1024) + 1024
  retry
end

# Start remote process with the value of port

socket = server.accept
puts socket.gets
socket.print "HTTP/1.1 200/OK\rContent-type: text/plain\r\n\r\n" + t
socket.close
```

This accomplishes (strong word) the same as the snippet in the question.

Share Improve this answer Follow

answered Oct 14, 2008 at 11:34

Marius Marais
532 ● 2 ● 6 ● 12

5 Just to nit-pick... Picking randomly like this is an unbounded loop, and can theoretically run for an undefined amount of time on a server; a situation that gets worse the more ports already in use. – Nigel Thorne Nov 27, 2012 at 20:54

1

You can try random-port, a simple Ruby gem (I'm the author):

```
require 'random-port'
```

```
port = RandomPort::Pool.new.aquire
```

The best way, though, is to release it afterward:

```
RandomPort::Pool::SINGLETON.new.acquire do |port|
  # Use the TCP port, it will be returned back
  # to the pool afterward.
end
```

The pool is thread-safe and it guarantees that the port won't be used by another thread or anywhere else in the app, until it's released.

Share
Improve this answer
Follow

edited Nov 6, 2021 at 19:07
rogerdpack
66.5k ● 39 ● 282 ● 401

answered Oct 4, 2018 at 6:48
yegor256
105k ● 130 ● 460 ● 620

0

Maybe you want test weather one port is listening, following is worked for me `system('(6<>/dev/tcp/127.0.0.1/9292) &>/dev/null')`, return true if 9292 is listening, otherwise, return false.

Share Improve this answer Follow

answered Aug 9, 2018 at 13:47
zw963
1,239 ● 16 ● 12

-8

Don't communicate on random ports. Pick a default one and make it configurable. Random ports are incompatible with firewalls. FTP does this and firewall support for it is a nightmare - it has to deeply inspect packets.

Share Improve this answer Follow

answered Oct 14, 2008 at 10:01
Tometzky
23.9k ● 5 ● 63 ● 78

7 The question was how to find a free port to use, not whether or not it is a good idea to use a random port. – ZombieDev Jun 13, 2013 at 14:08

1 Not all TCP/IP traffic flows through firewalls. – Roy Tinker Aug 25, 2016 at 20:45

I guess you could try all ports > 5000 (for example) in sequence. But how will you communicate to the client program what port you are listening to? It seems simpler to

-9

decide on a port, and then make it easily configurable, if you need to move your script between different enviroments.

For HTTP, the standard port is 80. Alternative ports i've seen used are 8080, 880 and 8000.

Share Improve this answer Follow

answered Oct 14, 2008 at 9:32

gnud

78.5k ● 5 ● 67 ● 79

See (new) last paragraph of question. – Marius Marais Oct 14, 2008 at 9:54

2 This should not be the accepted answer; see stackoverflow.com/a/201528/3528 below. – Rob Howard Feb 2, 2014 at 21:06