



Guides

RUBYGEMS BASICS

WHAT IS A GEM?

MAKE YOUR OWN GEM

GEMS WITH EXTENSIONS

NAME YOUR GEM

PUBLISHING YOUR GEM

SECURITY PRACTICES

MANAGING OWNERS USING UI

REMOVING A PUBLISHED GEM

SSL CERTIFICATE UPDATE

PATTERNS

SPECIFICATION REFERENCE

COMMAND REFERENCE

RUBYGEMS API

RUBYGEMS.ORG API

RUBYGEMS.ORG API V2.0

RUBYGEMS.ORG RATE LIMITS

API KEY SCOPES

RUN YOUR OWN GEM SERVER

SETTING UP MULTIFACTOR AUTHENTICATION

USING MFA IN COMMAND LINE

MFA REQUIREMENT OPT IN

USING S3 AS GEM SOURCE

RESOURCES

CONTRIBUTING TO RUBYGEMS

FREQUENTLY ASKED QUESTIONS

PLUGINS

CREDITS

COMMON VULNERABILITIES AND EXPOSURES

RELEASING RUBYGEMS

Need to serve gems locally or for your organization?

There are times you would like to run your own gem server. You may want to share gems with colleagues when you are both without internet connectivity. You may have private code, internal to your organization, that you'd like to distribute and manage as gems without making the source publicly available.

There are a few options to set up a server to host gems from within your organization. This guide covers the [Gemstash](#), [Gem in a Box](#), and [Gemirro](#) projects. It also discusses how to use these servers as gem sources during development.

RUNNING GEMSTASH

Gemstash is both a cache for remote servers (such as <https://rubygems.org>), and a private gem source.

To get started, install **gemstash**:

```
$ gem install gemstash
```

After it is installed, start the Gemstash server with the following command:

```
$ gemstash start
```

By default, the server runs on port 9292.

If you want to use it as a cache, you can tell Bundler to use Gemstash to find gems from RubyGems.org:

```
$ bundle config mirror.https://rubygems.org http://localhost:9292
```

With this configuration, all gems fetched from RubyGems.org via bundler are cached by Gemstash.

You can also push your own gems and use the gemstash server as a private gemsource. For more information about gemstash features and commands, read the [Gemstash](#) documentation.

RUNNING GEM IN A BOX

For a server with more features, including the ability to push gems, try out the [Gem in a Box](#) project.

To get started, install **geminabox**:

```
[~/dev/geminabox] gem install geminabox
```

Make a data directory for storing gems:

```
[~/dev/geminabox] mkdir data
```

Include the following in a **config.ru** file:

```
[~/dev/geminabox] cat config.ru  
require "rubygems"
```

```
require "geminabox"

Geminabox.data = "./data"
run Geminabox::Server
```

And run the server:

```
[~/dev/geminabox] rackup
[2011-05-19 12:09:40] INFO  WEBrick 1.3.1
[2011-05-19 12:09:40] INFO  ruby 1.9.2 (2011-02-18) [x86_64-darwin10.5]
[2011-05-19 12:09:40] INFO  WEBrick::HTTPServer#start: pid=60941 port=
```

Now you can push gems using the **gem inbox** command. The first time you do this, you'll be prompted for the location of your gem server.

```
[~/dev/secretgem] gem build secretgem.gemspec
Successfully built RubyGem
Name: secretgem
Version: 0.0.1
File: secretgem-0.0.1.gem
[~/dev/secretgem] gem inbox ./secretgem-0.0.1.gem
Enter the root url for your personal geminabox instance. (E.g. http://localhost:9292)
Host: http://localhost:9292
Pushing secretgem-0.0.1.gem to http://localhost:9292/...
```

There is a web interface available on <http://localhost:9292> as well. For more information, read the [Gem in a box](#) README.

RUNNING GEMIRRO

If you need a simple application that makes it easy way to create your own RubyGems mirror without having to push or write all gem you wanted in a

configuration file try out **Gemirro**. It does mirroring without any authentication and you can add your private gems in the gems directory. More, to mirroring a source, you only need to start the server, and gems will automatically be downloaded when needed.

To get started, install **gemirro**:

```
$ gem install gemirro
```

The process of setting up a mirror is fairly easy and can be done in few seconds.

The first step is to set up a new, empty mirror directory. This is done by running the **gemirro init** command.

```
$ gemirro init /srv/http/mirror.com/
```

Once created you can edit the main configuration file called **config.rb**. This configuration file specifies what source to mirror, destination directory, server host and port, etc.

Once configured and if you add gem in the **define_source**, you can pull them by running the following command:

```
$ gemirro update
```

Once all the Gems have been downloaded you'll need to generate an index of all the installed files. This can be done as following:

```
$ gemirro index
```

Last, launch the server, and all requests will check if gems are detected, and download them if necessary and generate index immediately.

```
$ gemirro server --start  
$ gemirro server --status  
$ gemirro server --restart  
$ gemirro server --stop
```

A web interface will be available at <http://localhost:2000>.

For more information, read the [Gemirro](#) README.

USING GEMS FROM YOUR SERVER

Whether you use Gemstash, Gem in a Box, Gemirro or another gem server, you can configure RubyGems to use your local or internal source alongside other sources such as <http://rubygems.org>.

Use the **gem sources** command to add the gem server to your system-wide gem sources. The following URL is the default for running Gem in a Box via **rackup**:

```
gem sources --add http://localhost:9292
```

Then install gems as usual:

```
[~] gem install secretgem
Successfully installed secretgem-0.0.1
1 gem installed
```

If you're using **Bundler** then you can specify this server as a gem source in your **Gemfile**:

```
[~/dev/myapp] cat Gemfile
source "http://localhost:9292"
gem "secretgem"

[~/dev/myapp] bundle
Using secretgem (0.0.1)
Using bundler (1.0.13)
Your bundle is complete! Use `bundle show [gemname]` to see where a b
```

[⏪ PREVIOUS](#)[NEXT ⏩](#)[STATUS](#)[UPTIME](#)[CODE](#)[DISCUSS](#)[STATS](#)[BLOG](#)[ABOUT](#)

HELP

RubyGems.org is the Ruby community's gem hosting service. Instantly publish your gems and install them. Use the API to interact and find out more information about available gems. Become a contributor and enhance the site with your own changes.

Supported by**Ruby Central***Supported by**Designed by***DOCKYARD***Resolved with***dnsimple***Optimized by**Monitored by***DATADOG***Monitored by* **Runscope***Gems served by***fastly** *Honeybadger**Monitored by*