

Complete and Fully Managed Hosting Environment for Web Applications

Rails Developers' Cookbook

March 2009

About this Cookbook

This cookbook is intended for Rails developers and provides technical information on how to deploy a Rails application to the Morph AppSpace. For more information about Morph Labs and its services, visit www.mor.ph.

Table of Contents

Deploying an Application to the Morph AppSpace	3
Preparing your Rails application	3
Deploying your Rails application	5
<i>Entering Morph Username and Password in the Command Line</i>	7
<i>Using Version Control Systems other than Subversion</i>	7
<i>Using 7zip for archiving on Windows</i>	8
<i>Using the Mounted Storage Feature</i>	9
Contact Morph Labs Customer Support	10

Deploying an Application to the Morph AppSpace

Deploying any Rails application on the Morph AppSpace is easy. Follow these steps:

Preparing your Rails application

To prepare your Rails application and reduce the likelihood of running into any hitches during deployment, read the following items:

- We recommend that you lock all your gems for production applications so that your application is using the gems you have tested and know work, rather than relying on the server gems which may or may not be compatible.



Note: For locking a gem into your vendor directory, visit these sites for the procedures:

- <http://gemsonrails.rubyforge.org/>
- <http://errtheblog.com/post/50-vendor-everything>

If one or more of your gems cannot be locked, send your gem installation request to support@mor.ph and our team will check if we can add it to our servers. For a list of pre-installed gems, see the **List of Installed Gems** at <http://forums.mor.ph/forums/8/topics/46>.

- Capistrano 2.x must be installed on your system. If you are using a version control system other than Subversion, you need to install Capistrano 2.3.0 on your system.
- Ensure that you have installed the Echoe 3 gem on your system.
- If your Rails application needs a database, make sure that it has migrations files under the db/migrate directory. We automatically call `rake db:migrate` during your application's deployment. For more information about Rails migrations, see <http://guides.rubyonrails.org/migrations.html>. For applications using the Rails 2.1 framework, visit <http://railscasts.com/episodes/107>.



If you would like to populate your database with pre-existing data, you may use the DB Admin in the Morph Control Panel to import the said data into your application's database. If you need further assistance, please do not hesitate to file a support ticket.

- Your application must be compatible with the PostgreSQL and MySQL database servers.
- Configure your application to use the database for sessions. You can create the migration for that by typing (in your rails application root) `rake db:sessions:create`. Then configure your `environment.rb` to use ActiveRecord based sessions and run `rake db:migrate`.



Note: Alternatively, you can use the Rails 2.x cookie-based sessions.

- Your machine must have a Tar archiving application. If you are using Windows and do not currently have Unix tools installed, please go to <http://gnuwin32.sourceforge.net/packages/libarchive.htm> and download the Tar archiving application.



Note: Make sure that the Tar archiving application is on the C:\ path. In the `C:\Program Files\GnuWin32\bin` folder, create a copy of `bsdtar` and rename it into `tar`.

- Any files to be uploaded by your application's users should be stored on a shared storage system such as Amazon S3 and not on the filesystem used by your application's code. This is because Morph AppSpace is a distributed system and a file stored in one cube will not be accessible on another. One way to achieve this is to use the `attachment_fu` plugin.
- We recommend that your code must be checked into a version control system supported by Capistrano 2.x. The deployment script defaults to using Subversion (SVN).
- Your latest code must be checked into your version control system (if you are using one) since the deployment script uses the code in your repository when deploying your application.



Note: You can use a version control system other than Subversion. You can use CVS, GIT, Mercurial, Accurev, Bazaar, Perforce, and Darcs. If you opt to deploy a locally stored Rails project, read the detailed procedure at <http://forums.mor.ph/forums/8/topics/25>.

- If you are using Git as your version control system, add a `.gitignore` file in `lib/` and then commit this to Git.
- If your application needs to send emails, we recommend that you use a dedicated email service provider.
- To setup the ActionMailer, follow the configurations below (in `config/environment.rb`):

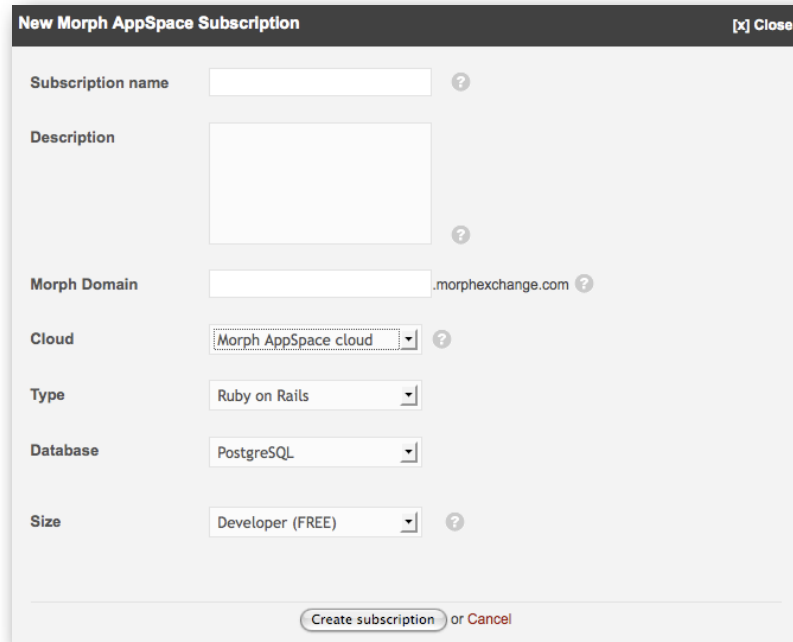
```
ActionMailer::Base.delivery_method = :smtp
ActionMailer::Base.smtp_settings = {
  :address => "your.smtp.server",
  :port => 25,
  :domain => "www.mywebsite.com",
  :user_name => 'your_user_if_login_is_needed',
  :password => 'your_password_if_login_required',
  :authentication => :login # Can also use: :plain or :cram_md5
}
```

For more configuration options, visit this site: <http://api.rubyonrails.org/>.

Deploying your Rails application

Follow these easy steps to deploy and run your Rails application on the Morph AppSpace:

1. Login to the Morph Control Panel.
2. From the Morph Control Panel **Subscriptions page**, create a new subscription by clicking **New Subscription**. This displays:



The screenshot shows a web form titled "New Morph AppSpace Subscription" with a close button "[x] Close" in the top right corner. The form contains the following fields:

- Subscription name:** A text input field with a help icon (?) to its right.
- Description:** A large text area with a help icon (?) to its right.
- Morph Domain:** A text input field containing ".morphexchange.com" with a help icon (?) to its right.
- Cloud:** A dropdown menu with "Morph AppSpace cloud" selected and a help icon (?) to its right.
- Type:** A dropdown menu with "Ruby on Rails" selected.
- Database:** A dropdown menu with "PostgreSQL" selected.
- Size:** A dropdown menu with "Developer (FREE)" selected and a help icon (?) to its right.

At the bottom of the form, there are two buttons: "Create subscription" and "or Cancel".

3. Fill in the required fields. If you have created a Morph AppCloud, you may choose to deploy your application on it.
4. Click **Create subscription** to create your new Morph AppSpace.
5. Select your newly added Morph AppSpace widget. If your application needs a database, click **Create Database**.

6. Download the **Deployer Script** file from the Morph AppSpace widget. Save this file to your application's root directory.

makepoems

Status: Stopped
Version: No code deployed

Size: [Developer (FREE)]
Database: Not created ([create now](#))

 **Deployer Script**

 **Manage**



Note: We recommend that you commit the Deployer Script to your version control system.

7. Use a text editor to open the Deployer Script. Replace the `set :repository` line with your repository path. If you are not using a version control system, follow the steps provided at <http://forums.mor.ph/forums/8/topics/25>.



Example: `set :repository, 'http://dev.myapp.com/repos/ruby/trunk/testing/makepoem'`

8. In the command console, go to your application's root directory and type `cap -f morph_deploy.rb morph:deploy`. If you saved the Deployer Script using a different filename or in another directory, replace `morph_deploy.rb` accordingly.
9. In the command console, you will be asked to type in your Morph login credentials. Type your username and password.
10. Once the command has finished, wait for a few minutes while your application is being initialized, and then in a browser go to `<your_app>.morphexchange.com`.

Entering Morph Username and Password in the Command Line

You can optionally enter your Morph username and password into the script or pass them to the command line. The variables are `morph_user` and `morph_pass`. You can either set these variables in the script or pass them as: `-s morph_user=username -s morph_pass=password`.

Using Version Control Systems other than Subversion

If you are using a version control system other than Subversion, perform these steps:

1. Click **Manage** on the appropriate Morph AppSpace widget. This displays the **Dashboard**.
2. Check the **Ruby Tools for Morph AppSpace** and click **Deployer Script Wizard**.
3. Select the version control system that you are using from the drop-down menu. You may choose any of the following:
 - Subversion
 - Git
 - Mercurial
 - Bazaar
 - Localhost
4. Enter the **Repository URL** in the field provided. This automatically saves your Repository URL in the Capistrano file.
5. Click **Generate** to proceed. This allows you to download the Capistrano file.
6. In the terminal, type `cap -f morph_deploy.rb morph:deploy` to deploy your application to the Morph AppSpace.
7. Otherwise, click **Cancel**.

Using 7zip for archiving on Windows

To deploy an application using 7zip for archiving on Windows, follow these steps:

1. Install 7zip on your machine. You can download it at <http://www.7-zip.org>.
2. Add 7zip on your Windows path. To do this, perform the following:
 - 2.1 Go to **My Computer>Properties>Advance System Settings**.
 - 2.2 Access **Advance Tab>Environment Variables**.
 - 2.3 In the System Variables Highlight "Path", click **Edit**.
 - 2.4 Type this line: C:\Program Files\7-zip.
3. Edit the morph_deploy.rb file. Perform the following:

Remove this command...
<pre>remove_files([morph_tmp_dir, 'code_update.tar.gz'])</pre>
...change to this.
<pre>remove_files([morph_tmp_dir, 'code_update.tar.gz', 'code_update.tar'])</pre>

Remove this command...
<pre>system("tar -C #{morph_tmp_dir} -czf code_update.tar.gz --exclude='./.*' .")</pre>
...change to this.
<pre>comp_dir="#{morph_tmp_dir}\\app\\.\\.*" system("7z a -ttar -r code_update.tar #{comp_dir}") system("7z a -tgzip code_update.tar.gz code_update.tar")</pre>

Remove this command...
<pre>flist = `tar tzf code_update.tar.gz` all_in = flist.include?('lib/') && flist.include?('app/') && flist.include?('config/environment.rb')</pre>
...change to this.
<pre>flist = `7z l code_update.tar` all_in = flist.include?('lib\\') && flist.include?('app\\') && flist.include?('config\\environment.rb')</pre>

Thanks to Lukas Franklin who provided Morph Labs with the solution in using 7zip for archiving during deployment.

Using the Mounted Storage Feature

Morph AppSpace supports the Mounted Storage feature for paid Morph AppSpaces for Rails. This feature provides Morph AppSpace paying subscribers a persistent storage for their applications' files.

The Mounted Storage is available in this path: `APP_DOC_ROOT + ../../shared/mounted_storage`



Example: `RAILS_ROOT + ../../shared/mounted_storage`

All files or folders, which include your application's uploaded images, videos, other files such as cached files, filesystem session-based, and so forth, saved under the Mounted Storage folder will persist among all computes.



Note: Rails application's public folder of any application deployed to a paid Morph AppSpace is automatically moved to the Mounted Storage folder. All uploaded files or cached files saved in the public folder are automatically persistent.

Contact Morph Labs Customer Support

For questions or comments, contact us at support@mor.ph. For more information, visit our online Forums at <http://forums.mor.ph>.



Morph Labs is the leading provider of Platform as a Service (PaaS) that virtualizes the application environment through the use of open source technologies to simplify the deployment, delivery, and management of Web-based applications.

Morph Labs uses virtual infrastructures including Amazon Web Services to provide a truly elastic environment for web applications that can be instantly provisioned and seamlessly scaled.

Morph Labs is a global company with headquarters in Cebu City, Philippines with additional in-country operations in Manila along with Los Angeles, California in the U.S.A.

www.mor.ph

World Headquarters

Unit 1A, Asiatown IT Park,
Lahug, Cebu City 6000
Philippines

Worldwide Inquiries

info@mor.ph
www.mor.ph

Copyright 2008, Morphlabs, Inc. All rights reserved.
This document is provided for information purposes only and the contents herein are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. Morph, Morph Application Platform, Morph AppSpace, and Simplify Innovation are registered trademarks of Morphlabs, Inc. Other names may be trademarks of their respective owners.