

# Installation Plan

## Overview

Steps based on this [guide](#) and this [guide](#).

## Steps (BlueHost specific):


1. Get BlueHost credentials
2. Create subdomain (here named scheduling)
3. Click Subdomain button in cPanel

The screenshot displays the BlueHost cPanel interface. On the left, there is a sidebar with a 'find' search bar, 'frequently accessed areas' (File Manager, Ruby on Rails), and a 'stats' section for the domain 'volunteerinnisfree.org'. The main area is divided into several sections: 'file management' (Site Backup Pro, File Manager, Legacy File Manager, FTP Manager, File Count, Disk Space Usage, Web Disk), 'database tools' (MySQL Databases, MySQL® DB Wizard, phpMyAdmin, Remote MySQL, MySQL Usage, Reset MySQL Password, PostgreSQL Databases, PostgreSQL DB Wizard, phpPgAdmin, Remote PostgreSQL), and 'domain management'. A red arrow points to the 'Subdomains' icon in the 'domain management' section. Below the 'domain management' section, there are icons for 'DNS Zone Editor' and 'CloudFlare'.

Storing a large number of files on your account can cause a degradation of performance. To

- 4.
5. Fill out subdomain information

## Subdomains

Subdomains are URLs for different sections of your website, which have an added *prefix* to your main domain name. Subdomains are relative to your accounts home directory. The  icon signifies your home directory.

### create a subdomain



For example, if your domain is volunteerinnisfree.org, a **sub-domain** of that might be **support.volunteerinnisfree.org**.

 .volunteerinnisfree.org

Home folder (aka Document Root) for your subdomain: /public\_html/

### Existing Subdomains

subdomains	root domain	document root	redirection	actions
main	.volunteerinnisfree.org	 /public_html/main	http://www.innisfreevi...	<input type="button" value="redirect"/> <input type="button" value="remove"/>
test scheduling	.volunteerinnisfree.org	 /public_html/testsc...	not redirected	<input type="button" value="redirect"/> <input type="button" value="remove"/>

- 
- 
- 
- 
- 
- 6.
7. Create Ruby on Rails application through cPanel
8. Click Ruby on Rails button in cPanel

Addon Domains	1 / ∞
FTP Accounts	2 / ∞
Mailing Lists	0 / ∞
All SQL Databases	2 / ∞
MySQL Databases	1
PostgreSQL Databases	1
Hosting package	Platinum Pak
Hostname	box263.bluehost.com
cPanel Version	11.42.0 (build 23)
Theme	bluehost
Apache version	2.2.29
PHP version	5.4.24
MySQL version	5.5.41-37.0-log
Architecture	x86_64
Operating system	linux
Shared IP Address	69.89.31.63
Path to sendmail	/usr/sbin/sendmail
Path to Perl	/usr/bin/perl
Perl version	5.10.1
Kernel version	3.12.32.1415814620
cPanel Pro	1.0 (RC1)

[collapse stats](#)

statistics

Choose Stats

Latest Visitors

Bandwidth

Awstats

Webalizer

Webalizer FTP

Access Logs

Error Logs

CPU Throttling

advanced

Apache Handlers

Image Manager

Index Manager

Error pages

Cron jobs

Server Status

Process Manager

404 Settings

MIME Types

software/services

CGI Center

Perl Modules

PHP Pear Packages

PHP Config

RubyGems

Ruby on Rails

preferences

Getting Started

Free Webinars

Change Password

Change Language

Shortcuts

How-To Videos

Renew Account

Update Billing Info



- 9.
10. Fill out application information

## Manage Ruby on Rails Applications

Ruby on Rails Applications are based on the Rails framework. Rails applications must be run like any other application. After creating an application, you will need to populate it with your code. Then, you can choose to start or stop the application and even to load the application every time the server reboots.

### Create Ruby on Rails Application

### Available Ruby on Rails Applications

APP NAME	PATH	PRODUCTION	DELETE
----------	------	------------	--------

**Please Note:** Bluehost provides Phusion Passenger. As of Ruby on Rails 3, running applications through fastcgi is no longer supported. We also recommend updating any applications that may be using Mongrel (or similar) so that they can take advantage of Phusion Passenger.

[← Go Back](#)

- 11.
12. Create a MySQL database for the scheduling application, and add a user to it
13. Using BlueHost's File Manager tool, copy zipped code to BlueHost (place it in the home directory for now)
14. Alternatively, you can skip this step and pull the code from GitHub later
15. SSH to BlueHost server using putty or a similar tool
16. Change to the rails\_apps directory: `cd ~/rails_apps`
17. Install Rails: `gem install rails -v 4.1.1 --no-rdoc --no-ri`
18. Install Rake: `gem install rake -v '10.4.2' --no-rdoc --no-ri`
19. Edit .bash.rc file as described in the first [guide](#)
20. Unzip application code: `"unzip ~/innisfree.zip"`
  - Type A when prompted to overwrite all common files
  - If you chose to instead pull the code from GitHub, do that now: `git clone https://github.com/uva-slp/innisfree`
21. `cd innisfree`
22. Uncomment line containing `gem 'therubyracer'` in Gemfile
23. Install missing gems `bundle install --path vendor/bundle`
24. Create **app** .htaccess file (e.g. `~/rails_app/innisfree/public/.htaccess`) as described in the first [guide](#)
  - Change the second to last line to `SetEnv GEM_HOME /home2/voluntf8/ruby/gems`
25. Delete the subdomain directory: `rm -rf ~/public_html/scheduling`

26. Link subdomain and application directory as described in the first [guide](#)
  - Create symbolic link from domain to app: `ln -s ~/rails_apps/innisfree/public ~/public_html/scheduling`
27. Create `config/database.yml` from `config/database.yml.template` (in the Rails application directory) so production entry points to BlueHost SQL server:

```
production:
  adapter: mysql2
  pool: 5
  port: 3306
  database: $DATABASE_NAME
  username: $USER_NAME
  password: $PASSWORD
  host: localhost
```

28. Run `rake db:migrate`
29. Delete the `application.css` file, if it exists: `rm ~/rails_app/innisfree/app/assets/stylesheets/application.css`

# Generic instructions

1. Install Apache2
2. Install Ruby (version 1.9.3 is what we deployed on, 2.1.5 has been tested to work)
3. Install Rails (version 4.1.1 is what we deployed on, 4.x should work, but has not been tested)
4. Install passenger: `gem install passenger`
5. Passenger is an web & application server that integrates nicely with Apache and Nginx to run Rails apps
6. Install the passenger module for apache: `passenger-install-apache2-module`
7. You may need to enable this module `sudo a2enmod passenger` and restart apache  
`sudo service apache2 restart`
8. Clone the application repository from Github
9. Follow steps 6.vi through 6.x from above (sorry about the mix of Arabic and Roman numerals...)
10. Link the public directory of the application to the directory Apache is serving out of:  
`ln -s ./public /opt/bitnami/apache2/htdocs/scheduling`
11. This is by default `*/install_dir*/apache2/htdocs/application` where *install\_dir* is the directory Apache was installed to. For bitnami, this looks to be `/opt/bitnami/apache2/htdocs`
12. Your application should now be accessible from the server's IP address (or DNS, if that's your thing)
- 13.

## Updating your installation

Once you've got your code up and running, you may make some changes and want to update your deployed code. 1. SSH to the server using putty or a similar tool 2. Change to the innisfree application directory: `cd ~/rails_apps/innisfree` 3. Pull from the GitHub hosted repository: `git pull` - This will prompt you for your GitHub username and password. Once you enter that it'll copy all the updates to your server 4. Run the following command to restart Passenger and update the code: `touch tmp/restart.txt`



# Sample files

## Sample config/database.yml

```
# SQLite version 3.x
#   gem install sqlite3
#
#   Ensure the SQLite 3 gem is defined in your Gemfile
#   gem 'sqlite3'
#
default: &default

development:
  adapter: mysql2
  pool: 5
  port: 3306
  database: $DEV_DATABASE_NAME
  username: $USER_NAME
  password: $PASSWORD
  host: localhost

# Warning: The database defined as "test" will be erased and
# re-generated from your development database when you run "rake".
# Do not set this db to the same as development or production.
test:
  adapter: sqlite3
  pool: 5
  timeout: 5000
  database: db/test.sqlite3

production:
  adapter: mysql2
  pool: 5
  port: 3306
  database: $DATABASE_NAME
  username: $USER_NAME
  password: $PASSWORD
  host: localhost
```

## Sample public/.htaccess file

```
<IfModule mod_passenger.c>
  Options -MultiViews
  PassengerResolveSymlinksInDocumentRoot on
  #Set this to whatever environment (development, production) you'll be runn
  RailsEnv production
  RackBaseURI /
  SetEnv GEM_HOME /home2/voluntf8/ruby/gems
</IfModule>
```