
opsworks_ruby Documentation

Release 1.14.0

Igor Rzegocki <igor@rzegocki.pl>

Mar 06, 2019

1	Getting Started	3
1.1	Super Quick Start	3
1.2	Quick Start	3
1.3	Full configuration	3
2	Troubleshooting	5
2.1	Deployment fails with errors regarding application code	5
2.2	SSL not working in legacy browsers	5
2.3	Some applications on my Layer deploys, some of them not	5
2.4	Environment variables fail to update on clean restart	5
3	Supported Technologies	7
4	Requirements	9
4.1	Cookbooks	9
4.2	Platform	9
5	Attributes	11
5.1	Stack attributes	11
5.2	Cross-application attributes	12
5.3	Application attributes	12
5.4	Logrotate Attributes	24
6	Recipes	25
7	Contributing	27
7.1	Pull Requests	27
7.2	Running Tests	27
8	Author and Contributors	29
8.1	Author	29
8.2	Contributors	29
9	License	31

opsworks_ruby is a set of chef recipes to ease the deployment to AWS OpsWorks service. It was created, when Amazon introduced Chef 12 to OpsWorks stack without support for any chef recipes. The main goal of this project is to mimic Chef 11 OpsWorks stack as close as possible.

The code is open source, and available on [github](#).

The main documentation for the site is organized into a couple sections:

- *User Documentation*
- *Cookbook Documentation*
- *About opsworks_ruby*

1.1 Super Quick Start

This cookbook is designed to “just work”. So in base case scenario, all you have to do is create a layer and application with an optional assigned RDS data source, then add *Recipes* to the corresponding OpsWorks actions. The base scenario includes **nginx** as a web server, **puma** as application server, **Ruby on Rails** as framework, and no extra workers (sidekiq/resque/etc.). If this okay for you, and if you configured RDS and application properly from OpsWorks panel - you don't have to do anything else with those recipes.

1.2 Quick Start

Alternatively, you can use a *JSON configurator*, which is a nice tool which would help you quickly generate Stack/Layer JSON (and hassle free). It supports, all the configuration parameters in most of the cases. If you wish to configure multiple apps, use the generator for each one of them separately.

1.3 Full configuration

You can always create your JSON file manually, just head to the *Attributes* section of this documentation, and check which one you need.

2.1 Deployment fails with errors regarding application code

If you have enabled `app['framework']['envs_in_console']` parameter, always disable it first and check if error occurs again. This parameter is a “hack” on rails `config/application.rb` file, which inserts all application environment variables before initializing the Rails stack. This may cause some problems (in very rare cases), and break this file.

2.2 SSL not working in legacy browsers

By default webserver is configured to follow strict SSL security standards, [covered in this article](#). However, old browsers (like IE < 9 or Android < 2.2) wouldn't work with this configuration very well. If your application needs a support for those browsers, set `app['webserver']['ssl_for_legacy_browsers']` to true.

2.3 Some applications on my Layer deploys, some of them not

Check the `node['applications']` parameter in *Attributes* section. If set, it narrows down the list of applications allowed to deploy, to its value. If not sure what to do - try to remove from your Stack/Layer config and see if this helps.

2.4 Environment variables fail to update on clean restart

If you changed `after_deploy` method in your appserver, to do a [clean restart](#), please note that ENV variables, are not refreshed from applicatino perspective. More details are covered in [Issue #142](#).

Supported Technologies

- Database
 - MariaDB
 - MySQL
 - PostgreSQL
 - Sqlite3
- Sources
 - Amazon S3
 - Git
- Framework
 - Null (no framework)
 - hanami.rb
 - Padrino
 - Ruby on Rails

App server

- Null (no appserver)
- Puma
- Thin
- Unicorn

Web server

- Null (no webservers)
- Apache2
- nginx

Worker

- Null (no worker)
- delayed_job
- resque
- sidekiq
- shoryuken

4.1 Cookbooks

- `deployer`
- `ruby-ng`
- `nginx`

4.2 Platform

This cookbook was tested on the following OpsWorks platforms:

- Amazon Linux 2017.03
- Ubuntu 16.04 LTS

In addition, all recent Debian family distributions are assumed to work.

Attributes format follows the guidelines of old Chef 11.x based OpsWorks stack.

So all of them, need to be placed under `node['deploy'][<application_shortcode>]`.

Attributes (and whole logic of this cookbook) are divided to six sections.

Following convention is used: `app == node['deploy'][<application_shortcode>]`

so for example `app['framework']['adapter']` actually means

`node['deploy'][<application_shortcode>]['framework']['adapter']`.

5.1 Stack attributes

These attributes are used on Stack/Layer level globally to configure the `opsworks_ruby` cookbook itself. They should't be used under `node['deploy'][<application_shortcode>]` (notice lack of the `app[]` convention).

- `node['applications']`
- An array of application shortnames which should be deployed to given layer. If set, only applications with `deploy` flag set (on OpsWorks side) included in this list will be deployed. If not set, all `deploy` application will be supported. This parameter mostly matters during the setup phase, since all application in given stack are deployed to the given layer. Using this parameter you can narrow the list to application which you actually intend to use. **Important** thing is, that when you try to do a manual deploy from OpsWorks of an application, not included in this list - it will be skipped, as this list takes precedence over anything else.
- `node['ruby-version']`
- **Type:** string
- **Default:** 2.6
- Sets the Ruby version used through the system. For debian-based distributions, a `ruby-ng` cookbook is used (check [ruby-ng cookbook documentation](#)). For Amazon Linux, packages provided by distribution (i.e. `ruby23`, `ruby23-devel` etc.). **Important** please note, that some versions may be available on one system, and not on the other (for example `ruby-ng` gets freshest versions of ruby way earlier than Amazon Linux).

5.2 Cross-application attributes

These attributes can only be set at the server level; they cannot vary from application to application.

5.2.1 webservice

- `node['defaults']['webservice']['remove_default_sites']`
- **Type:** array
- **Default:** `%w[default default.conf 000-default 000-default.conf default-ssl default-ssl.conf]`
- **Note:** Only applies to Apache2 webservice
- A list of “default site” filenames that should be removed (if they exist) from `/etc/{httpd,apache2}/sites-enabled` in order to disable any “default site” provided by the OS-provided Apache2 package. Set this to `nil` or an empty array (`[]`) if you want the default site to be enabled.

5.3 Application attributes

5.3.1 global

Global parameters apply to the whole application, and can be used by any section (framework, appserver etc.).

- `app['global']['environment']`
- **Type:** string
- **Default:** `production`
- Sets the “deploy environment” for all the app-related (for example `RAILS_ENV` in Rails) actions in the project (server, worker, etc.)
- `app['global']['deploy_dir']`
- **Type:** string
- **Default:** `/srv/www/app_name`
- Determines where the application will be deployed.
- Note that if you override this setting, you’ll typically want to include the `short_name` in the setting. In other words, this setting doesn’t override the `/srv/www` base directory default; it overrides the application-specific `/srv/www/app_name` default.
- `app['global']['symlinks']`
 - **Type:** key-value
 - **Default:** `{ "system": "public/system", "assets": "public/assets", "cache": "tmp/cache", "pids": "tmp/pids", "log": "log" }`
 - **Important Notice:** Any values for this parameter will be *merged* to the defaults
 - List of symlinks created to the shared directory. The format is `{"shared_path": "release_path"}`. For example `{"system", "public/system"}` means: Link `/src/www/app_name/current/public/system` to `/src/www/app_name/shared/system`.
- `app['global']['create_dirs_before_symlink']`

- **Type:** array
- **Default:** ["tmp", "public", "config", "../..shared/cache", "../..shared/assets"]
- **Important Notice:** Any values for this parameter will be *appended* to the defaults
- List of directories to be created before symlinking. Paths are relative to `release_path`. For example `tmp` becomes `/srv/www/app_name/current/tmp`.
- `app['global']['purge_before_symlink']`
 - **Type:** array
 - **Default:** ["log", "tmp/cache", "tmp/pids", "public/system", "public/assets"]
 - **Important Notice:** Any values for this parameter will be *appended* to the defaults
 - List of directories to be wiped out before symlinking. Paths are relative to `release_path`. For example `tmp` becomes `/srv/www/app_name/current/tmp`.
- `app['global']['rollback_on_error']`
 - **Type:** boolean
 - **Default:** true
 - When set to true, any failed deploy will be removed from `releases` directory.
- `app['global']['logrotate_rotate']`
 - **Type:** integer
 - **Default:** 30
 - **Important Notice:** The parameter is in days
 - How many days of logfiles are kept.
 - See Logrotate Attributes for more information on logrotate attribute precedence.
- `app['global']['logrotate_frequency']`
 - **Type:** string
 - **Default:** daily
 - **Supported values:** daily, weekly, monthly, size X
 - How often logrotate runs for the given log(s), either time-based or when the log(s) reach a certain size.
 - See Logrotate Attributes for more information on logrotate attribute precedence.
- `app['global']['logrotate_options']`
 - **Type:** Array
 - **Default:** %w[missingok compress delaycompress notifempty copytruncate sharedscripts]
 - All of the unqualified options (i.e., without arguments) that should be enabled for the specified logrotate configuration.
 - See Logrotate Attributes for more information on logrotate attribute precedence.
- `app['global']['logrotate_X']`
 - **Type:** Varies

- Any attribute value Y for `logrotate_X` will cause the `[logrotate_app](https://github.com/stevendanna/logrotate/blob/master/resources/app.rb)` resource `_X_` to be called with argument Y. For example setting `logrotate_cookbook` to `'my_cookbook'` will result in the `logrotate_app` resource being invoked with the resource value `cookbook 'my_cookbook'`.
- See Logrotate Attributes for more information on logrotate attribute precedence.

5.3.2 database

Those parameters will be passed without any alteration to the `database.yml` file. Keep in mind, that if you have RDS connected to your OpsWorks application, you don't need to use them. The chef will do all the job, and determine them for you.

- `app['database']['adapter']`
- **Supported values:** `aurora`, `aurora-postgresql`, `mariadb`, `mysql`, `postgis`, `postgresql`, `sqlite3`, `null`
- **Default:** `sqlite3`
- ActiveRecord adapter which will be used for database connection. `null` means that no database will be configured, and is currently only tested with the `rails` framework.
- `app['database']['username']`
- Username used to authenticate to the DB
- `app['database']['password']`
- Password used to authenticate to the DB
- `app['database']['host']`
- Database host
- `app['database']['database']`
- Database name
- `app['database'] [<any other>]`
- Any other key-value pair provided here, will be passed directly to the `database.yml`

5.3.3 source

Those parameters can also be determined from OpsWorks application, and usually you don't need to provide them here.

- `app['source']['adapter']`
- **Supported values:** `git`, `http`, `s3`
- **Default:** `git`
- Source used by the cookbook to fetch the application codebase.
- `app['source']['url']`
- Source code URL (repository URL for SCMs).

git

- `app['source']['remove_scm_files']`
- **Supported values:** `true`, `false`
- **Default:** `true`
- If set to `true`, all SCM leftovers (like `.git`) will be removed.
- `app['source']['revision']`
- Branch name/SHA1 of commit which should be use as a base of the deployment.
- `app['source']['ssh_key']`
- A private SSH deploy key (the key itself, not the file name), used when fetching repositories via SSH.
- `app['source']['ssh_wrapper']`
- A wrapper script, which will be used by git when fetching repository via SSH. Essentially, a value of `GIT_SSH` environment variable. This cookbook provides one of those scripts for you, so you shouldn't alter this variable unless you know what you're doing.
- `app['source']['generated_ssh_wrapper']`
- **Default:** `/tmp/ssh-git-wrapper.sh`
- If the cookbook generates an SSH wrapper for you, this is where it will generate it. For users whose `/tmp` partitions are mounted `noexec` (a good security practice to prevent code injection exploits), this attribute allows you to override that location to a partition where execution of the generated shell script is allowed.
- `app['source']['enable_submodules']`
- If set to `true`, any submodules included in the repository, will also be fetched.

s3

This source expects a packed project in one of the following formats:

`bzip2`, `compress`, `gzip`, `tar`, `xz` or `zip`.

If you are using `ubuntu`, `7zip` is also supported.

- `app['source']['user']`
- `AWS_ACCESS_KEY_ID` with read access to the bucket.
- `app['source']['password']`
- `AWS_SECRET_ACCESS_KEY` for given `AWS_ACCESS_KEY_ID`.

http

This source expects a packed project in one of the following formats:

`bzip2`, `compress`, `gzip`, `tar`, `xz` or `zip`.

If you are using `ubuntu`, `7zip` is also supported.

- `app['source']['user']`
- If file is hidden behind HTTP BASIC AUTH, this field should contain username.

- `app['source']['password']`
- If file is hidden behind HTTP BASIC AUTH, this field should contain password.

5.3.4 framework

Pre-optimization for specific frameworks (like migrations, cache etc.).

Currently `hanami.rb` and `Rails` are supported.

- `app['framework']['adapter']`
- **Supported values:** `null`, `hanami`, `padrino`, `rails`
- **Default:** `rails`
- Ruby framework used in project.
- `app['framework']['migrate']`
- **Supported values:** `true`, `false`
- **Default:** `true`
- If set to `true`, migrations will be launch during deployment.
- `app['framework']['migration_command']`
- A command which will be invoked to perform migration. This cookbook comes with predefined migration commands, well suited for the task, and usually you don't have to change this parameter.
- `app['framework']['assets_precompile']`
- **Supported values:** `true`, `false`
- **Default:** `true`
- `app['framework']['assets_precompilation_command']`
- A command which will be invoked to precompile assets.
- `app['framework']['logrotate_name']`
- **Type:** `string`
- **Default:** Depends on adapter-specific behaviors
- The name of the `logrotate_app` resource, and generated configuration file, for the specified app framework logrotate configuration.
- Unlike other logrotate attributes, this attribute can only be set or overridden at a the app framework level; there are no app-wide or global settings beyond those provided by the framework library
- `app['framework']['logrotate_log_paths']`
 - **Type:** `Array`
 - **Default:** Depends on adapter-specific behaviors
 - Which log file(s) should be backed up via logrotate. If this parameter evaluates to an empty array, no logs will be backed up for the specified app framework.
 - Unlike other logrotate attributes, this attribute can only be set or overridden at a the app framework level; there are no app-wide or global settings beyond those provided by the framework library.

padrino

For Padrino, slight adjustments needs to be made. Since there are many database adapters supported, instead of creating configuration for each one, the `DATABASE_URL` environmental variable is provided. You need to parse it in your `config/database.rb` file and properly pass to the configuration options. For example, for ActiveRecord:

```
database_url = ENV['DATABASE_URL'] &&_
↳ActiveRecord::ConnectionAdapters::ConnectionSpecification::ConnectionUrlResolver.
↳new(ENV['DATABASE_URL']).to_hash
ActiveRecord::Base.configurations[:production] = database_url || {
  :adapter => 'sqlite3',
  :database => Padrino.root('db', 'dummy_app_production.db')
}
```

rails

- `app['framework']['envs_in_console']`
- **Supported values:** `true, false`
- **Default:** `false`
- If set to `true`, `rails console` will be invoked with all application-level environment variables set.
- **WARNING!** This is highly unstable feature. If you experience any troubles with deployments, and have this feature enabled, consider disabling it as a first step in your debugging process.

5.3.5 appserver

Configuration parameters for the ruby application server. Currently Puma, Thin, Unicorn, and Passenger are supported.

- `app['appserver']['adapter']`
- **Default:** `puma`
- **Supported values:** `puma, thin, unicorn, passenger, null`
- Server on the application side, which will receive requests from webserver in front. `null` means no appserver enabled.
- `app['appserver']['application_yaml']`
- **Supported values:** `true, false`
- **Default:** `false`
- Creates a `config/application.yml` file with all pre-configured environment variables. Useful for gems like `figaro`
- `app['appserver']['dot_env']`
- **Supported values:** `true, false`
- **Default:** `false`

- Creates a `.env` file with all pre-configured environment variables. Useful for gems like `dotenv`
- `app['appserver']['preload_app']`
- **Supported values:** `true`, `false`
- **Default:** `true`
- Enabling this preloads an application before forking worker processes.
- `app['appserver']['timeout']`
- **Default:** `50`
- Sets the timeout of worker processes to seconds.
- `app['appserver']['worker_processes']`
- **Default:** `4`
- Sets the current number of worker processes. Each worker process will serve exactly one client at a time.
- `app['appserver']['passenger_version']`
- **Default:** `None`
- Which Debian APT package version should be installed from the PPA repo provided by Passenger. Currently this defaults to the latest version provided by the Passenger APT PPA. Set this to a non-nil value to lock your Passenger installation at a specific version.
- `app['appserver']['after_deploy']`
 - **Default:** `stop-start`
 - **Supported values:** `stop-start`, `restart`, `clean-restart`
 - Tell the appserver how to restart following a deployment. A `stop-start` will instruct the appserver to stop and then start immediately. This can cause requests from the webserver to be dropped since it closes the socket. A `restart` sends a signal to the appserver instructing it to restart while maintaining the open socket. Requests will hang while the app boots, but will not be lost. A `clean-restart` will perform a `stop-start` if the Gemfile has changed or a `restart` otherwise. The behavior of each of these approaches varies between appservers. See their documentation for more details.
- `app['appserver']['port']`
 - **Default:** `None`
 - Bind the appserver to a port on `0.0.0.0`. This is useful for serving the application directly from the appserver without a web server middleware or separating the web server into its own container or server.

unicorn

- `app['appserver']['backlog']`
- **Default:** `1024`
- `app['appserver']['delay']`
- **Default:** `0.5`
- `app['appserver']['tcp_nodelay']`
- **Supported values:** `true`, `false`
- **Default:** `true`
- `app['appserver']['tcp_nopush']`

- **Supported values:** true, false
- **Default:** false
- `app['appserver']['tries']`
- **Default:** 5

puma

- `app['appserver']['log_requests']`
- **Supported values:** true, false
- **Default:** false
- `app['appserver']['thread_max']`
- **Default:** 16
- `app['appserver']['thread_min']`
- **Default:** 0
- `app['appserver']['on_restart']`
- Code to run before doing a restart. This code should close log files, database connections, etc.
- `app['appserver']['before_fork']`
- Code to run immediately before the master starts workers.
- `app['appserver']['on_worker_boot']`
- Code to run in a worker before it starts serving requests. This is called everytime a worker is to be started.
- `app['appserver']['on_worker_shutdown']`
- Code to run in a worker right before it exits. This is called everytime a worker is to about to shutdown.
- `app['appserver']['on_worker_fork']`
- Code to run in the master right before a worker is started. The worker's index is passed as an argument. This is called everytime a worker is to be started.
- `app['appserver']['after_worker_fork']`
- Code to run in the master after a worker has been started. The worker's index is passed as an argument. This is called everytime a worker is to be started.

thin

- `app['appserver']['max_connections']`
- **Default:** 1024
- `app['appserver']['max_persistent_connections']`
- **Default:** 512
- `app['appserver']['timeout']`
- **Default:** 60
- `app['appserver']['worker_processes']`
- **Default:** 4

passenger

- `app['appserver']['max_pool_size']`
- **Type:** Integer
- **Default:** Passenger-provided default (based on server capacity)
- Sets the `PassengerMaxPoolSize` parameter
- `app['appserver']['min_instances']`
- **Type:** Integer
- **Default:** Passenger-provided default (based on server capacity)
- Sets the `PassengerMinInstances` parameter
- `app['appserver']['mount_point']`
- **Default:** /
- Which URL path should be handled by Passenger. This option allows you to configure your application to handle only a subset of requests made to your web server. Useful for certain hybrid static/dynamic web sites.

5.3.6 webserver

Webserver configuration. Proxy passing to application is handled out-of-the-box.

Currently Apache2 and nginx is supported.

- `app['webserver']['adapter']`
- **Default:** `nginx`
- **Supported values:** `apache2, nginx, null`
- Webserver in front of the instance. It runs on port 80 by default (see `app['webserver']['port']`), and receives all requests from the Load Balancer/Internet. `null` means no webserver enabled.
- `app['webserver']['dhparams']`
- If you wish to use custom generated DH primes, instead of common ones (which is a very good practice), put the contents (not file name) of the `dhparams.pem` file into this attribute. [Read more here](#).
- `app['webserver']['keepalive_timeout']`
- **Default:** `15`
- The number of seconds webserver will wait for a subsequent request before closing the connection.
- `app['webserver']['ssl_for_legacy_browsers']`
- **Supported values:** `true, false`
- **Default:** `false`
- By default webserver is configured to follow strict SSL security standards, [covered in this article](#). However, old browsers (like IE < 9 or Android < 2.2) wouldn't work with this configuration very well. If your application needs a support for those browsers, set this parameter to `true`.
- `app['webserver']['port']`
- **Default** `80`
- The port on which the webserver should listen for HTTP requests.

- `app['webserver']['ssl_port']`
- **Default** 443
- The port on which the webserver should listen for HTTPs requests, if SSL requests are enabled. Note that SSL itself is controlled by the `app['enable_ssl']` setting in Opsworks.
- `app['webserver']['force_ssl']`
- **Supported values:** `true`, `false`
- **Default** `false`
- When this parameter is set to `true` all requests passed to `http` will be redirected to `https`, with 301 status code. This works only when SSL in OpsWorks panel is enabled, otherwise it's omitted.
- `app['webserver']['site_config_template']`
- **Default** `appserver.apache2.conf.erb` or `appserver.nginx.conf.erb`
- The name of the cookbook template that should be used to generate per-app configuration stanzas (known as a "site" in apache and nginx configuration parlance). Useful in situations where inserting an `extra_config` text section doesn't provide enough flexibility to customize your per-app webserver configuration stanza to your liking.
- Note that when you use a custom site configuration template, you can also choose to define `extra_config` as any data structure (e.g., Hash or even nested Hash) to be interpreted by your custom template. This provides somewhat unlimited flexibility to configure the webserver app configuration however you see fit.
- `app['webserver']['site_config_template_cookbook']`
- **Default** `opsworks_ruby`
- The name of the cookbook in which the site configuration template can be found. If you override `app['webserver']['site_config_template']` to use a site configuration template from your own cookbook, you'll need to override this setting as well to ensure that the `opsworks_ruby` cookbook looks for the specified template in your cookbook.
- `app['webserver']['logrotate_name']`
- **Type:** string
- **Default:** Depends on adapter-specific behaviors
- The name of the `logrotate_app` resource, and generated configuration file, for the specified app webserver logrotate configuration.
- Unlike other logrotate attributes, this attribute can only be set or overridden at a the app webserver level; there are no app-wide or global settings beyond those provided by the webserver library
- `app['webserver']['logrotate_log_paths']`
 - **Type:** Array
 - **Default:** Depends on adapter-specific behaviors
 - Which log file(s) should be backed up via logrotate. If this parameter evaluates to an empty array, no logs will be backed up for the specified app webserver.
 - Unlike other logrotate attributes, this attribute can only be set or overridden at a the app webserver level; there are no app-wide or global settings beyond those provided by the webserver library

apache

- `app['webserver']['extra_config']`
- Raw Apache2 configuration, which will be inserted into `<Virtualhost *:80>` (or other port, if specified) section of the application.
- `app['webserver']['extra_config_ssl']`
- Raw Apache2 configuration, which will be inserted into `<Virtualhost *:443>` (or other port, if specified for SSL) section of the application. If set to `true`, the `extra_config` will be copied.
- `app['webserver']['limit_request_body']`
- **Default:** 1048576
- `app['webserver']['log_level']`
- **Default:** info
- `app['webserver']['log_dir']`
- **Default:** /var/log/apache2 (debian) or /var/log/httpd (rhel)
- A place to store application-related Apache2 logs.
- `app['webserver']['proxy_timeout']`
- **Default:** 60

nginx

- `app['webserver']['build_type']`
- **Supported values:** default or source
- **Default:** default
- The way the `nginx` cookbook handles `nginx` installation. Check out the [corresponding docs](#) for more details. Never use `node['nginx']['install_method']`, as it will be always overwritten by this attribute.
- `app['webserver']['client_body_timeout']`
- **Default:** 12
- `app['webserver']['client_header_timeout']`
- **Default:** 12
- `app['webserver']['client_max_body_size']`
- **Default:** 10m
- `app['webserver']['extra_config']`
- Raw `nginx` configuration, which will be inserted into `server` section of the application for HTTP port.
- `app['webserver']['extra_config_ssl']`
- Raw `nginx` configuration, which will be inserted into `server` section of the application for HTTPS port. If set to `true`, the `extra_config` will be copied.
- `app['webserver']['log_dir']`
- **Default:** /var/log/nginx
- A place to store application-related `nginx` logs.

- `app['webserver']['proxy_read_timeout']`
- **Default:** 60
- `app['webserver']['proxy_send_timeout']`
- **Default:** 60
- `app['webserver']['send_timeout']`
- **Default:** 10
- `app['webserver']['enable_upgrade_method']`
- **Supported values:** true, false
- **Default:** false
- When set to true, enable Websocket's upgrade method such as Rails actionCable.

Since this driver is basically a wrapper for [nginx cookbook](#), you can also configure `node['nginx']` attributes as well (notice that `node['deploy'] [<application_shortname>]` logic doesn't apply here.)

5.3.7 worker

sidekiq

- `app['worker']['config']`
- Configuration parameters which will be directly passed to the worker. For example, for sidekiq they will be serialized to `sidekiq.yml` config file.

delayed_job

- `app['worker']['queues']`
- Array of queues which should be processed by delayed_job

resque

- `app['worker']['workers']`
- **Default:** 2
- Number of resque workers
- `app['worker']['queues']`
- **Default:** *
- Array of queues which should be processed by resque

shoryuken

- `app['worker']['config']`
 - Configuration parameters which will be directly passed to the worker. For example, for shoryuken they will be serialized to the relevant `shoryuken.yml config file`.
- `app['worker']['process_count']`
 - **Default:** 1
 - Number of shoryuken runner daemons to start. Shoryuken is multithreaded, so defaults to 1.
- `app['worker']['require']`
 - Path to require, relative to the currently deployed application directory.
- `app['worker']['require_rails']`
 - Boolean: emits `-R` to require the rails environment on boot.
- `app['worker']['syslog']`
 - Boolean: configures piping shoryuken runner log output to syslog via `logger`

5.4 Logrotate Attributes

Logrotate behaviors occur across multiple drivers, for example webserver and framework. For this reason, the evaluation order for attribute-driven behaviors is a bit more complex for logrotate than for other options that are either entirely global (for example, `global.environment`) or entirely isolated to a single type of driver (`webserver.keepalive_timeout`).

The evaluation rules for logrotate setting `_X_` are as follows, from highest priority to lowest priority:

- `app[driver_type]['logrotate_X']`
- `app['global']['logrotate_X']`
- `node['defaults'][driver_type]['logrotate_X']`
- `node['defaults']['global']['logrotate_X']`

CHAPTER 6

Recipes

This cookbook provides five main recipes, which should be attached to corresponding OpsWorks actions.

- `opsworks_ruby::setup` - attach to **Setup**
- `opsworks_ruby::configure` - attach to **Configure**
- `opsworks_ruby::deploy` - attach to **Deploy**
- `opsworks_ruby::undeploy` - attach to **Undeploy**
- `opsworks_ruby::shutdown` - attach to **Shutdown**

Contributions are **welcome** and will be fully **credited**.

We accept contributions via Pull Requests on [Github](#).

7.1 Pull Requests

- **Lint your code** - The easiest way to apply the conventions is to install [overcommit](#).
- **Add tests!** - Your patch won't be accepted if it doesn't have tests.
- **Document any change in behaviour** - Make sure the README and any other relevant documentation are kept up-to-date.
- **Consider our release cycle** - We try to follow [semver](#). Randomly breaking public APIs is not an option.
- **Create topic branches** - Don't ask us to pull from your master branch.
- **One pull request per feature** - If you want to do more than one thing, send multiple pull requests.
- **Send coherent history** - Make sure each individual commit in your pull request is meaningful. If you had to make multiple intermediate commits while developing, please squash them before submitting.
- **Commit convention** - We follow [AngularJS Git Commit Message Conventions](#)

7.2 Running Tests

Ensure, that you have [Chef DK](#) installed.

```
bundle exec overcommit -r
chef exec rspec
```

Happy coding!

8.1 Author

Igor Rzegocki (@ajgon)

8.2 Contributors

- Nick Marden (@nickmarden)
- Phong Si (@phongsi)
- Kevin Olbrich (@olbrich)
- Kevin Pheasey (@kpheasey)
- Nathan Flood (@npflood)
- Teruo Adachi (@interu)
- Marcos Beirigo (@marcosbeirigo)
- John Calvin Young (@johncalvinyoung)
- Rich Seviora (@richseviora)

The MIT License (MIT)

Copyright (c) 2016-2019 Igor Rzegocki

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.