
Shelly Cloud

Painless Ruby deployment

Giedrius Rimkus
2012 July

Cloud Computing?

Pay-per-use

On-demand

Computing resources

Alternatives?

EngineYard <http://www.engineyard.com>

Heroku <http://www.heroku.com>

Scalarium <http://www.scalarium.com>

others..

What Shelly Cloud is?

Delivering platform as a service

Deployments are done by pushing with git

Example of git push

```
giedrius@rubylt:~/Projects/store (master) $ git push
gstore master
Counting objects: 81, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (72/72), done.
Writing objects: 100% (81/81), 27.62 KiB, done.
Total 81 (delta 6), reused 0 (delta 0)
remote:
remote: ---> Received push to cloud 'gstore'
remote: ---> Checking Gemfile
remote: ---> Push accepted
remote: ---> Start your cloud using: `shelly start --
cloud gstore`
remote:
To git@git.shellycloud.com:gstore.git
* [new branch]      master -> master
```

Aim

Helping developers

Developers focus on building application and not on deployment nor infrastructure

Requirements

- Ruby versions
 - MRI Ruby (1.9.3)
 - MRI Ruby (1.9.2)
 - Ruby Enterprise Edition (1.8.7)
 - Bundler
 - "shelly-dependencies" gem (optional)
 - Rails 3.x support
 - Rails 2.3.x support
 - Rack-based apps
 - Gems with native extensions
-

Shelly Cloud

- Running Ruby applications
- It doesn't have to be Rails application
 - Sinatra
 - or any other which is compatible with Rack interface

full list of supported frameworks
<https://github.com/rack/rack#supported-web-frameworks>

Shelly Gem

A tool for managing clouds on Shelly Cloud from command line.

Cloudfile

Your cloud management place

You can edit settings of your Cloud by making changes to the **Cloudfile**, committing them to the git repository and pushing those changes to the shelly git remote.

Features

- Running periodic tasks
 - Background processing
 - Storing Files
 - Caching (frontend, backend)
 - Sending emails
 - SSL support
 - Websockets support
-

Running periodic tasks

Cron job is a periodic task
Whenever

Modify Cloudfire

1. Add whenever to Gemfile

```
gem "whenever"
```

2. Run bundle

3. \$ bundle exec wheneverize .

4. define cronjobs in config/schedule.rb

```
env :MAILTO, "email@example.com"
```

Background processing

Delayed_jobs gem is used for this task.

Set up one of your apps for delayed_jobs

Add it to your Gemfile

Add jobs*

Commit and push (deploy)

*

https://github.com/collectiveidea/delayed_job/blob/master

Storing files

Kind of sharing files between instances and deployments

For example
uploads/

Caching

Frontend caching using Varnish

Backend caching using Memcache.

- cache is cleared after every deploy.

Sending emails

Sending emails works out of the box
SPF record (trusted mails)
1 mail server IP

SSL Support

When using domain `code-name.shellyapp.com` you get free SSL.

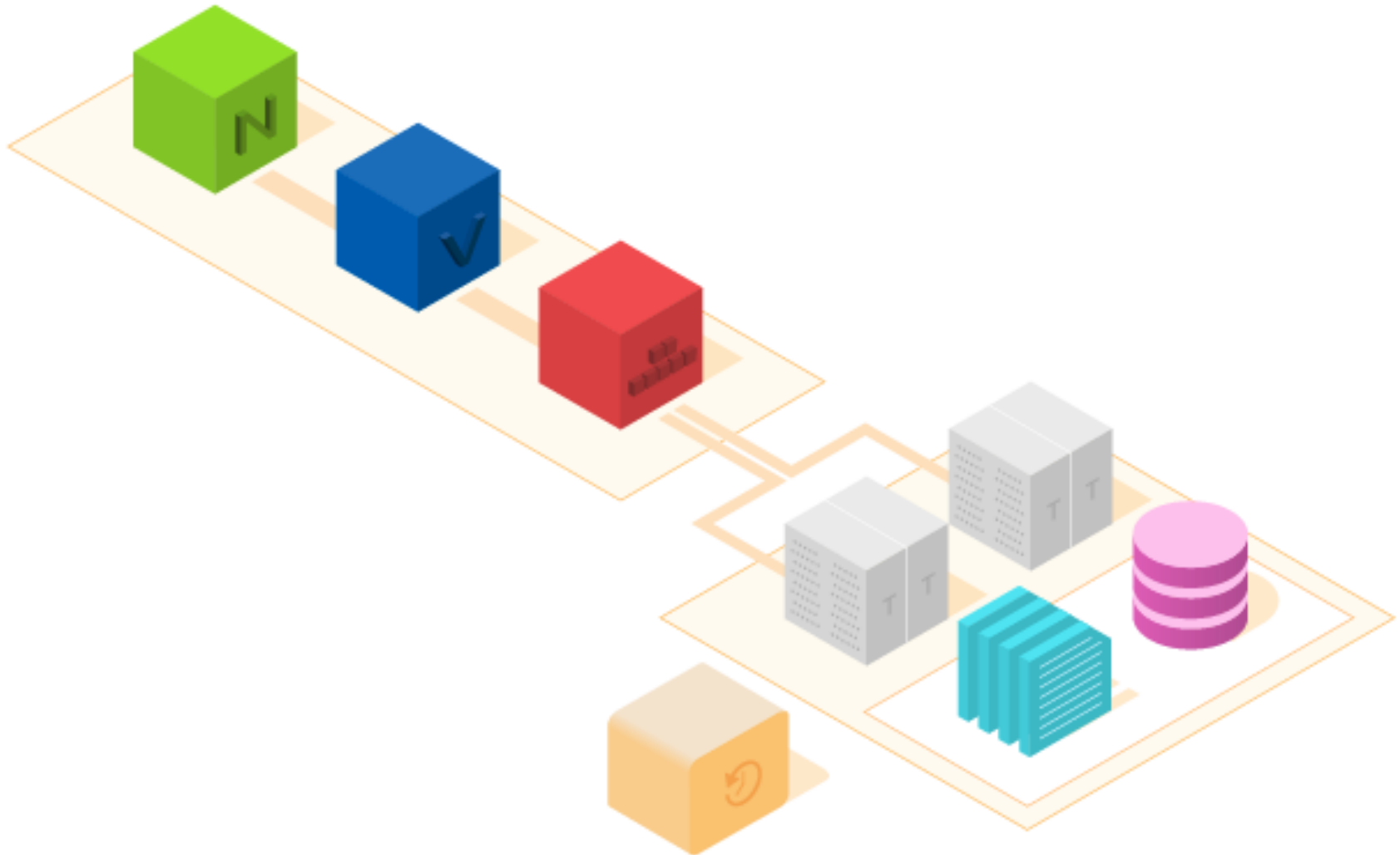
If you want own SSL, you will have to purchase SSL certificate from any CA as well as private IP from Shelly Cloud.

Websockets

You can run a separate thin server on any of your servers that will handle websockets traffic.

Create a `websockets.ru` file in the root directory of your project. Use this file to configure the application handling websockets traffic.

Architecture



Supported Databases

PostgreSQL

MongoDB

Redis

https://shellycloud.com/documentation/supported_databases

Shelly Console

You can use the shelly gem to run rails console or IRB for rack applications.

Console is ran on random application server with user privileges.

To run console, use `shelly console`

Price comparison / Small app

3 thin workers, 1 delayed_job, 1GB database

Heroku

\$107(workers) + \$15 (shared database) = \$122 ~ €100

EngineYard

\$90 (small instance, 1 CPU, 1.7GB of RAM) ~ €73

(no backups included)

Shelly Cloud

€40(1 large instance, 2 CPUs, 2GB of RAM) + €0.60 (1 GB DB storage) = €40.6

Price comparison / Medium app

10 workers, 3 delayed_job, 10GB database (dedicated)

Heroku

\$431 (workers) + \$250 dedicated database = \$681 ~ €558

Shelly Cloud

€160 (4 large instances for workers, 2 GB RAM, 2 CPUs) + €40 (1 large instance dedicated db) + €6 = €206

In total 10 GB of RAM, 10 virtual CPUs

EngineYard (Similar in case of RAM to shelly setup)

\$444 (2 medium inst 3.75GB of RAM (each), 2 CPUs, 1 small 1 CPU, 1.7GB of RAM),

\$24 (storage and bandwidth) = \$468 ~ €384

Let's do it in practice

LIVE deployment on Shelly Cloud

Redeem code for meetup attendees

Code: ARM2012

Valid for: **1 week**

Questions?

Thank You

Giedrius Rimkus

Software Engineer

@giedriusr
