

The Dark Ages of App Configuration

presented by Scott Motte (Mot)



Secrets lived in code
(2008)

cloud application platform
deploy and scale powerful apps



Forget Servers

Run Anything

See Everything

Trust & Manage

Then came Heroku (2009)

Deploy Ruby, Node.js, Clojure, and Java apps.

Get up and running in minutes, and deploy instantly with git.

Focus 100% on your code, and never think about servers,
instances, or VMs again.

Agile Deployment on Heroku ▶

```
$ heroku create
Creating sushi.herokuapp.com | git@heroku.com:sushi.git

$ git push heroku master
----> Heroku receiving push
----> Rails app detected
----> Compiled slug size is 8.0MB
----> Launching... done, v1
http://sushi.herokuapp.com deployed to Heroku
```

[How it Works](#)

It's free to get started and sign up is instant.

[Sign Up](#)

Procfile (2009)

```
1 Procfile
1 web: RAILS_ENV=production rails server
2 worker: RAILS_ENV=production script/delayed_job start
3
```

2 2 2 2 2 2 2 2 2 2 2 2 2

NORMAL

master

Procfile

33%

1:1

Foreman (2010)

 ddollar update readme ce5c8b4 · 15 years ago 

[Preview](#) [Code](#) [Blame](#) 96 lines (61 loc) · 2.58 KB     

foreman(1) -- manage Procfile-based applications

SYNOPSIS

```
foreman start [process]
foreman export <var>format</var> [location]
```

DESCRIPTION

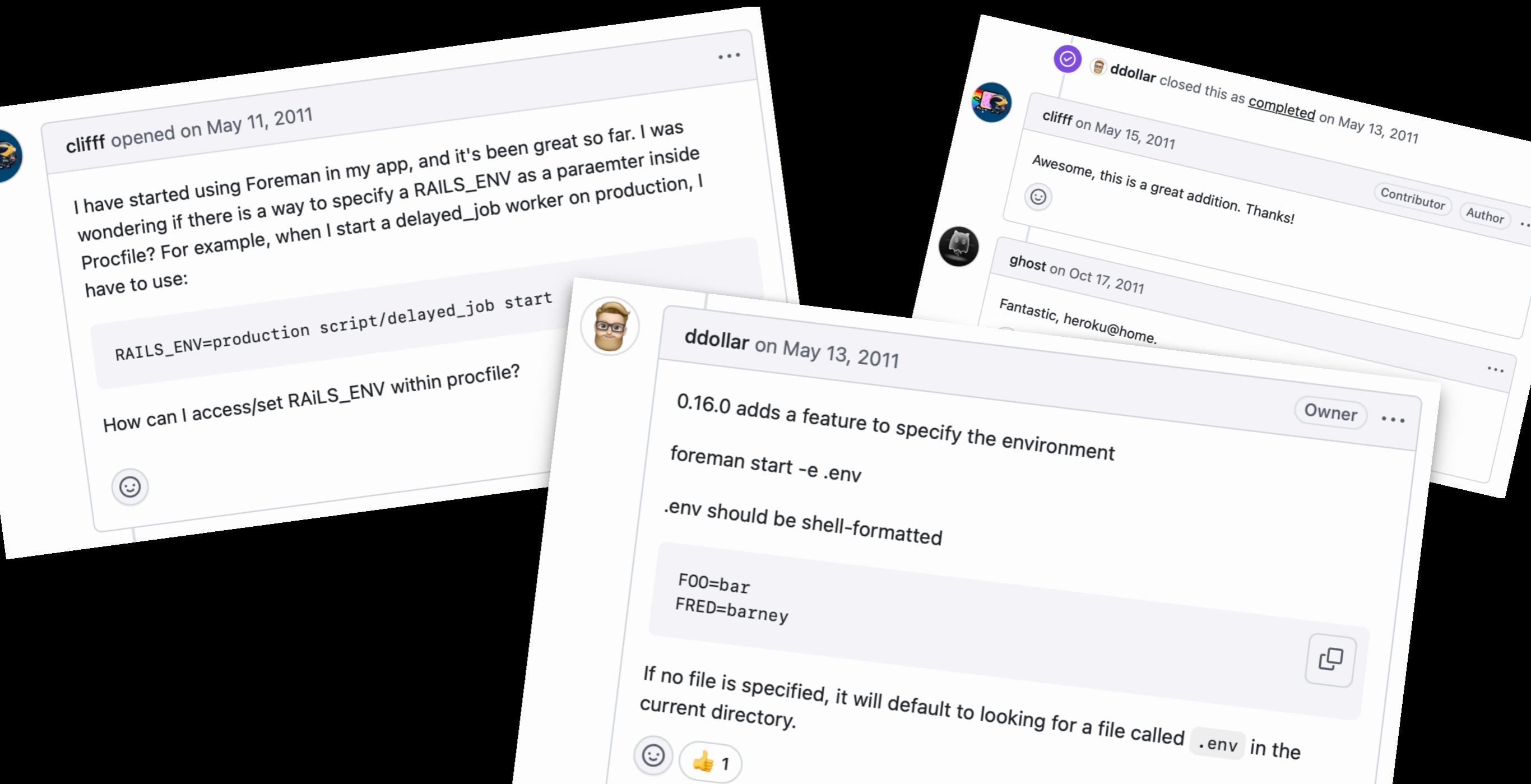
Foreman is a manager for Procfile-based applications. Its aim is to abstract away the details of the Procfile format, and allow you to either run your application directly or export it to some other process management format.

RUNNING

`foreman start` is used to run your application directly from the command line.

If no additional parameters are passed, foreman will run one instance of each type of process defined in your Procfile.

The Rise of .env Files



Twelve-Factor App (2011)



The screenshot shows a dark-themed web browser window. The address bar displays the URL "12factor.net/config" with a lock icon indicating it's secure. The main content area features a large white diamond logo in the center. To the right of the logo are three links: "Blog", "Community", and a blue "GitHub" button with a white octocat icon. Below the logo, the text "THE TWELVE-FACTOR APP" is displayed in a large, bold, white sans-serif font. The overall aesthetic is clean and minimalist.

III. Config

Store config in the environment

An app's *config* is everything that is likely to vary between deploys (staging, production, developer environments, etc). This includes:

- Resource handles to the database, Memcached, and other backing services
- Credentials to external services such as Amazon S3 or Twitter
- Per-deploy values such as the canonical hostname for the deploy

Dotenv is Born (2012)

The screenshot shows a GitHub commit history for the file `lib/dotenv/environment.rb`. The commit message indicates 13 files changed with +188 -0 lines changed. The code itself is a new file creation, starting with a module `Dotenv` and a class `Environment < Hash`. It includes an `initialize` method that takes a filename, sets it as `@filename`, and loads the file. The `load` method uses `read.each do |line|` to process each line. A specific line in the `load` block is highlighted with a blue border, showing the assignment of a value to `self[$1]`.

```
... @@ -0,0 +1,22 @@
1 + module Dotenv
2 +   class Environment < Hash
3 +     def initialize(filename)
4 +       @filename = filename
5 +       load
6 +     end
7 +
8 +     def load
9 +       read.each do |line|
10+         self[$1] = $2 if line =~ /\A([\w_]+)=(.*)\z/
11+       end
12+     end
13+
14+     def read
15+       File.read(@filename).split("\n")
16+     end
17+   end
18+ end
```

Ruby to Node (2013)

2013: phpdotenv

2013: python-dotenv

2013: dotenv (nodejs)

2013: godotenv

2013: dotenv_elixir

2014: dotenv-rs

2014: docker-compose

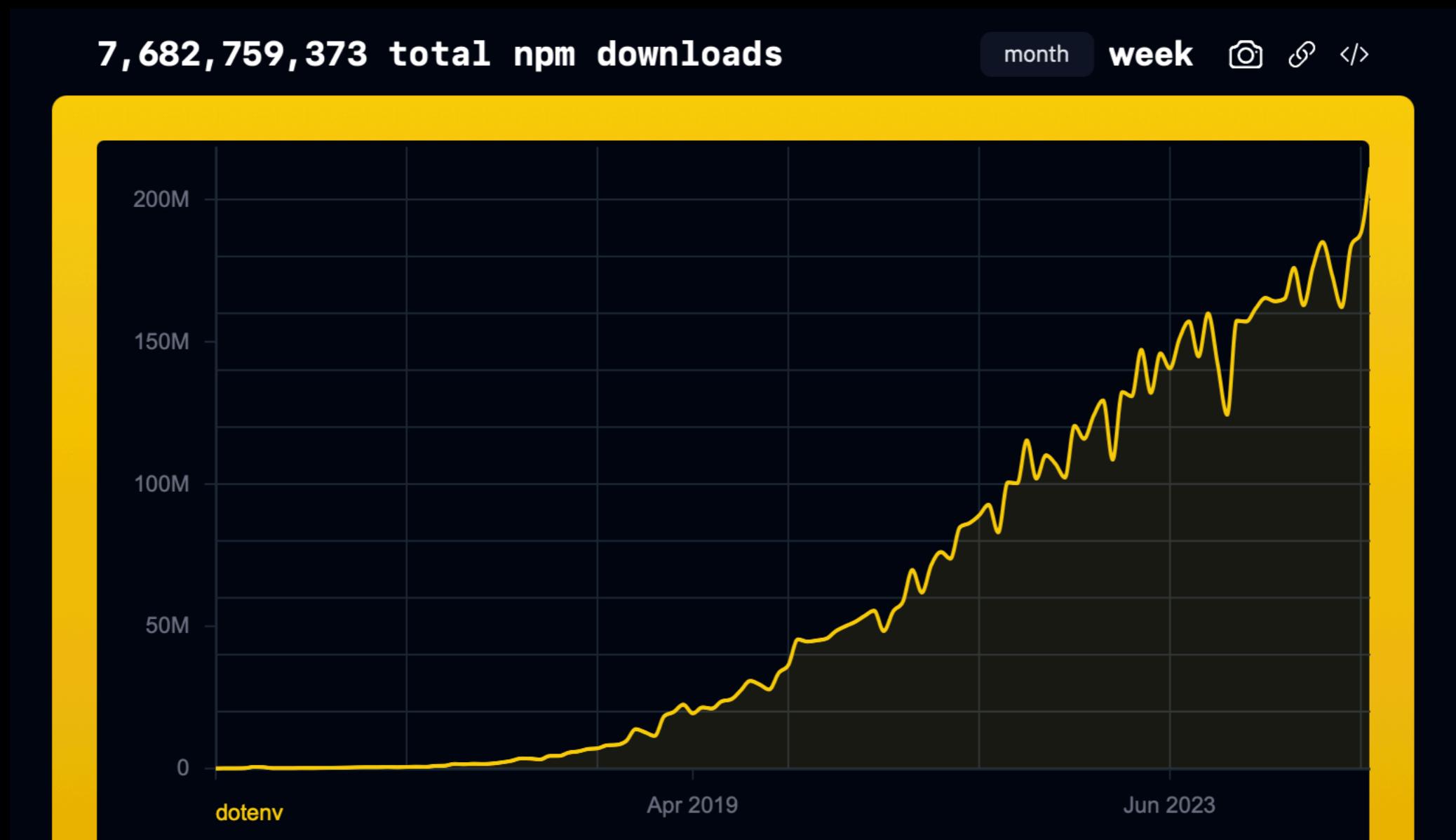
2017: dotenv.net

2017: dotenv-kotlin

2020: dotenv-java

2021: swift-dotenv

Small Code, Big Reach



Present Day – Rethinking .env

Three Problems

- Inconsistency across platforms
- Juggling multiple environments
- Leaking your .env file

Three Solutions

- Make it run the same everywhere
- Add first-class support for multiple environments
- Encrypt your .env files to counteract leaks

Introducing dotenvvx

The screenshot shows a web browser window with the URL dotenvvx.com in the address bar. The page content is as follows:

- .ENV Docs** (links)
- Features** **Pro** (links)
- dotenv.better.** (large title)
- a better dotenv—from the creator of* [dotenv](#) ★ 19.7k (GitHub link)
- `curl -fsS https://dotenvvx.sh | sh` (installation command in a box)
- [curl](#) [brew](#) [Documentation](#) (links)
- FEATURED #1 ON** **Hacker News** 354 (Hacker News badge)
- FEATURED #2 ON** **GitHub Trending** (GitHub Trending badge)
- [Read the Whitepaper](#) (link)

Run Anywhere

The screenshot shows a GitHub README page for the `dotenvx` repository. The page title is `dotenv / README.md`. The main content section is titled **Run Anywhere**. It includes a list of supported languages:

- ▾ Ruby 💎

For Ruby, it provides the following code example:

```
$ echo "HELLO=World" > .env
$ echo 'puts "Hello #{ENV["HELLO"]}"' > index.rb

$ dotenvx run -- ruby index.rb
Hello World
```

Below the code, there is a link to the [extended ruby guide](#).

Other supported languages listed include:

- ▶ Go 🐹
- ▶ Rust 🦀
- ▶ Java ☕

demo: dotenvx run --

Run Anywhere

Consistency

The screenshot shows a web browser window with the URL dotenvx.com/spec/. The page title is "Spec". The main content features a large heading "Dotenv Spec" and a subtitle: "A formal comparison of dotenv parsing behavior across major libraries, languages, and frameworks." Below this is a table comparing four tools across three test cases.

	dotenvx	docker	docker-compose	npm@dotenv	n
Pass Rate	100%	13%	63%	90%	
101_BASIC	✓	✓	✓	✓	
102_EMPTY	✓	✓	✓	✓	
103_MACHINE	✓	✓	✓	✓	

Multiple Environments

The screenshot shows a GitHub README page for the `dotenvx` repository. The title is `Multiple Environments`. Below it, a note says: "Create a `.env.production` file and use `-f` to load it. It's straightforward, yet flexible." A code block shows the following terminal session:

```
$ echo "HELLO=production" > .env.production
$ echo "console.log('Hello ' + process.env.HELLO)" > index.js

$ dotenvx run -f .env.production -- node index.js
[dotenvx@1.X.X] injecting env (1) from .env.production
Hello production
> ^^
```

At the bottom, there is a "More examples" section with three bullet points:

- ► multiple `'.env'` files
- ► `--overload` flag
- ► `--verbose` flag

```
demo: dotenvx run -f .env.production --
```

Prevent leaks

Encryption

The screenshot shows a GitHub README page for the `dotenvx` repository. The title is `dotenvx / README.md`. The page contains the following content:

Encryption

Add encryption to your `.env` files with a single command. Use `dotenvx encrypt`.

```
$ dotenvx encrypt
✓ encrypted (.env)
```

A terminal window below shows the contents of an encrypted `.env` file:

```
vim .env ~/C/d/p/app — vim
```

```
#-----[DOTENV_PUBLIC_KEY]-----
#/      public-key encryption for .env files      /
#/      [how it works](https://dotenvx.com/encryption)      /
#-----/
DOTENV_PUBLIC_KEY="03f8b376234c4f2f0445f392a12e80f3a84b4b0d1e0c3df85c4

# Database configuration
DB_HOST="encrypted:BNr24F4vW9CQ37LOXeRgOL6Q1wtJfAoAVXtSdSfpicPDHtqo/Q2
```

demo: dotenvx encrypt

Additional Benefits

- Support for write-only access
- Set a single environment variable on your infra
- PR Reviews for secrets

Additional Features

- ext gitignore
- ext precommit
- ext prebuild
- get
- built-in debugging and help

Go Deeper

The screenshot shows a web browser window with the URL dotenvx.com in the address bar. The main content area has a dark background with white text. A large blue button labeled "Read the Whitepaper" is prominently displayed. Below it, a section titled "Dotenvx: Reducing Secrets Risk with Cryptographic Separation" contains a detailed abstract and introduction. To the right, a white rectangular box displays a preview of a whitepaper titled "Dotenvx: Reducing Secrets Risk with Cryptographic Separation" by Scott Motte. The whitepaper preview includes the abstract, introduction, and secrets sections, along with a note at the bottom.

Read the Whitepaper

Dotenvx: Reducing Secrets Risk with Cryptographic Separation

Abstract. An ideal secrets solution would not only centralize secrets but also contain the fallout of a breach. While secrets managers offer centralized storage and distribution, their design creates a large blast radius, risking exposure of thousands or even millions of secrets. We propose a solution that reduces the blast radius by splitting secrets management into two distinct components: an encrypted secrets file and a separate decryption key.

Dotenvx: Reducing Secrets Risk with Cryptographic Separation

Scott Motte – DRAFT
mott@dotenvx.com
www.dotenvx.com

Abstract. An ideal secrets solution would not only centralize secrets but also contain the fallout of a breach. While secrets managers offer centralized storage and distribution, their design creates a large blast radius, risking exposure of thousands or even millions of secrets. We propose a solution that reduces the blast radius by splitting secrets management into two distinct components: an encrypted secrets file and a separate decryption key.

1. Introduction

Modern software relies on secrets to operate—API keys, tokens, and credentials are essential for applications to interact with services like Stripe, Twilio, and AWS. The majority of these secrets are stored in platform-native secrets managers such as AWS Secrets Manager, Vercel Environment Variables, and Heroku Config Vars. These systems offer convenience by centralizing secrets and seamlessly injecting them into runtime environments. However, this centralization introduces significant risks. If breached, they expose all secrets stored within, resulting in a blast radius where thousands or even millions of secrets may be leaked. At the same time, alternatives such as `.env` files minimize blast radius but lack the safeguards necessary to prevent unauthorized access. Developers are left choosing between simplicity with higher risk or complexity with a larger blast radius.

What is needed is a new system based on hybrid cryptography instead of trust, allowing a developer to encrypt secrets without relying on any third party to remain secure. In this paper, we propose a solution to these risks using a library that decrypts an encrypted secrets file at runtime with a private key stored separately in the platform's secrets manager. This approach contains the blast radius of a breach while maintaining the simplicity of `.env` files. Even if one component—either the encrypted file or the secrets manager—is compromised, secrets remain secure. Only simultaneous access to both can expose them.

2. Secrets

We define a secret as a token or string value, typically issued by a third-party

Thank you.

dotenvx.com

Sources

- Apr 24, 2019. Heroku Launches. https://blog.heroku.com/commercial_launch
- May 16, 2010. First Foreman commit. <https://github.com/ddollar/foreman/commit/2dbd3e6be5f5a45ee3b236735130beaedc436bb2>
- May 13, 2011. .env file introduced to Foreman. <https://github.com/ddollar/foreman/issues/17>, <https://github.com/ddollar/foreman/commit/9193a675a3e53739f412d4e493ab74594d1e826c#diff-d3b801ad9a2c2e0606e87eadd12d18b740274ba85bd0354a6ff749245e4d1deeR204>
- Jun 03, 2011. Twelve-Factor App is written. <https://github.com/adamwiggins/12factor/commit/2b06e7deabb64bb759f9fc6f4d9b6fcc546921bb>
- Jul 23, 2012. dotenv ruby is born. <https://github.com/bkeepers/dotenv/commit/c3568a06b341f1182bd4e8b0d6e58a594cac7966#diff-b335630551682c19a781afebcf4d07bf978fb1f8ac04c6bf87428ed5106870f5R5>
- Jan 22, 2013. phpdotenv is born. <https://github.com/vlucas/phpdotenv/commits/master/?after=20d6a1bdfd62910da28b447b2ccb39ac542e96b2+570>
- Jun 13, 2013. python-dotenv is born. <https://github.com/theskumar/python-dotenv/commit/5fc02b7303e8854243970e12564f2433da7a1f7f>
- Jul 5, 2013. dotenv node is born. <https://github.com/motdotla/dotenv/commit/71dabbf27b699fcb7a04714709cefc6e78892b9>

Sources continued

- Jul, 2013. godotenv is born. <https://github.com/joho/godotenv/commit/973cf53008332ef58d44121143d3ef29758c3352>
- Dec 03, 2013. dotenv_elixir is born. https://github.com/avdi/dotenv_elixir/commit/9a9d61ae4e449dc5f4fb414bd189183e339d7210
- Oct 23, 2014. dotenv-rs is born. <https://github.com/dotenv-rs/dotenv/commit/47570b116e1b9653be66861191152e7ad0a9078a>
- Dec 09, 2014. docker-compose. <https://github.com/docker/compose/pull/665>
- Nov 21, 2017. dotenv.net is born. <https://github.com/bolorundurowb/dotenv.net/commit/67714bc46294008aa6726323bf0319ad9e03c6d9>
- Nov 25, 2017. dotenv-kotlin is born. <https://github.com/cdimascio/dotenv-kotlin/commit/429a5172a8bc113818cf8687c6a15bd4244d9d35>
- Sep 18, 2020. dotenv-java is born. <https://github.com/cdimascio/dotenv-java/commit/4057ea1c1d930d74b5282f48d68332d517eca718>
- Oct 17, 2021. swift-dotenv is born. <https://github.com/thebarndog/swift-dotenv/commits/develop/?after=bed53315bc88e8e9d3a7eabb2b9440c6e9c7aa51+100>