StriderCD Book

Release 1.4

Niall O'Higgins

Sep 08, 2017

Contents

1 Introduction **3**
- 1.1 What Is Strider . . . 3
- 1.2 What Is Continuous Integration . . . 3
- 1.3 What Is Continuous Deployment . . . 3
- 1.4 Strider Philosophy . . . 4
- 1.5 Features . . . 4

2 Installing StriderCD **11**
- 2.1 Dependencies . . . 11
- 2.2 Latest Stable Version . . . 11
- 2.3 Latest Development Version . . . 11
- 2.4 Adding Users . . . 12
- 2.5 Cloud-hosted or Remote MongoDB . . . 12

3 Running Strider **13**

4 Creating a project **15**

5 Hacking Strider **17**
- 5.1 Auth . . . 17
- 5.2 CORS . . . 17

6 Indices and tables **19**

Contents:

CHAPTER 1

Introduction

What Is Strider

Strider is an Open Source Continuous Integration and Deployment platform. It is written in JavaScript/Node.JS and uses MongoDB. It is released under the BSD license. While similar conceptually to systems such as Travis and Jenkins, Strider is designed to be easy to setup, use, and customize.

What Is Continuous Integration

Continuous Integration (a.k.a. CI) is a software engineering process. It can be defined as running an automated test suite on every commit to a software project and notifying on success and failure.

CI is beneficial as it greatly increases the feedback developers receive on each commit, and the health and well-being of the software overall. It is amazing how many bugs can be found which otherwise would remain covered due to particulars of development environments.

The faster your team gets feedback on bugs and regressions, the sooner your release is ready!

What Is Continuous Deployment

At a high level, Continuous Deployment (a.k.a. CD) extends Continuous Integration to include automated deployment. This can vary from full "deploy-on-green" whereby the system is automatically deployed to production every time the tests pass, to more conservative automatic deployment to staging with a manual push to production.

Continuous Deployment typically includes a notion of "feature flippers" which enable new features to be rolled out gradually, to a subset of customers, or quickly toggled off if they cause issues.

CD makes the software release feedback loop as tight as possible.

Strider Philosophy

We believe that Continuous Integration and Continuous Deployment processes improve the quality and reliability of software.

We want to make these processes more accessible, and one of the main blockers is well-designed, high quality CI and CD tooling.

By choosing sane defaults, automating to the extent possible, and minimizing extraneous UI, Strider can elegantly meet the needs of 80+% users out of the box.

However we realise that in the real-world, integration with other systems and customization is often required. For this reason, Strider has been designed from the ground up to be as easy as reasonably possible to customize and extend - in very powerful ways.

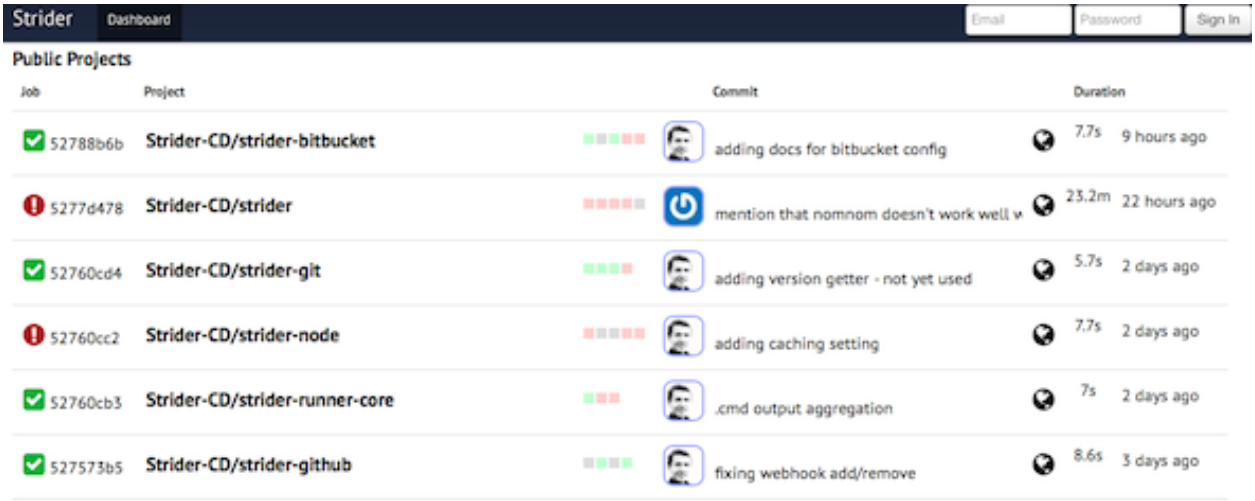
Strider has robust support for extensions (which are simply NPM modules). Have a complex workflow to trigger builds and deployments? Write a Job plugin to handle it. Use an unsupported VCS? Write a Provider plugin.

Features

Strider has the following major features:

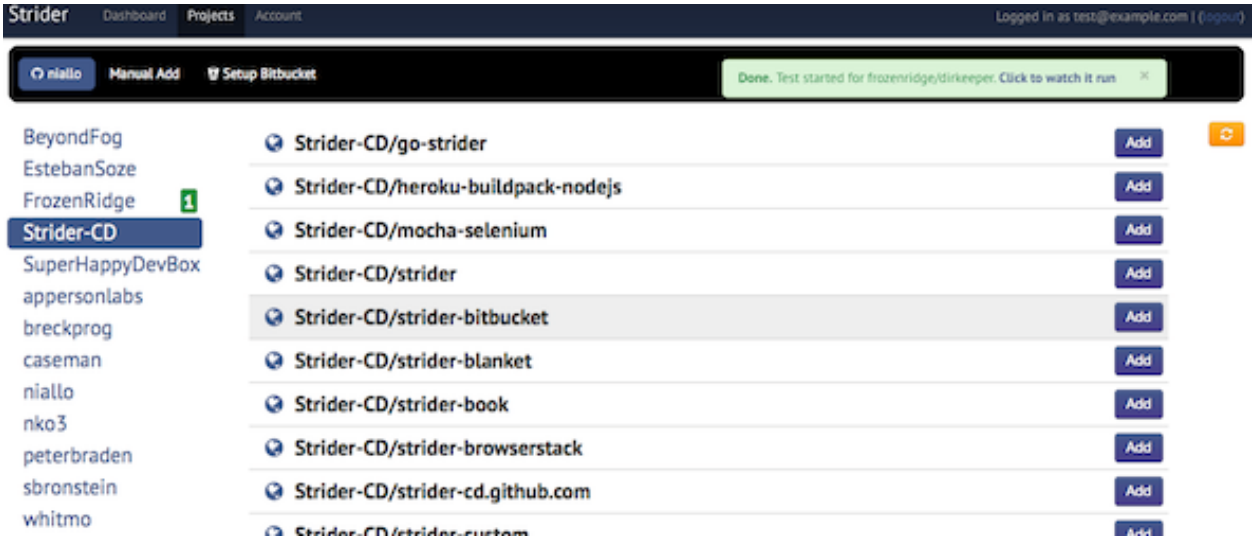
Dashboard

Stylish dashboard displaying the current and recent test and deploy status of each project:



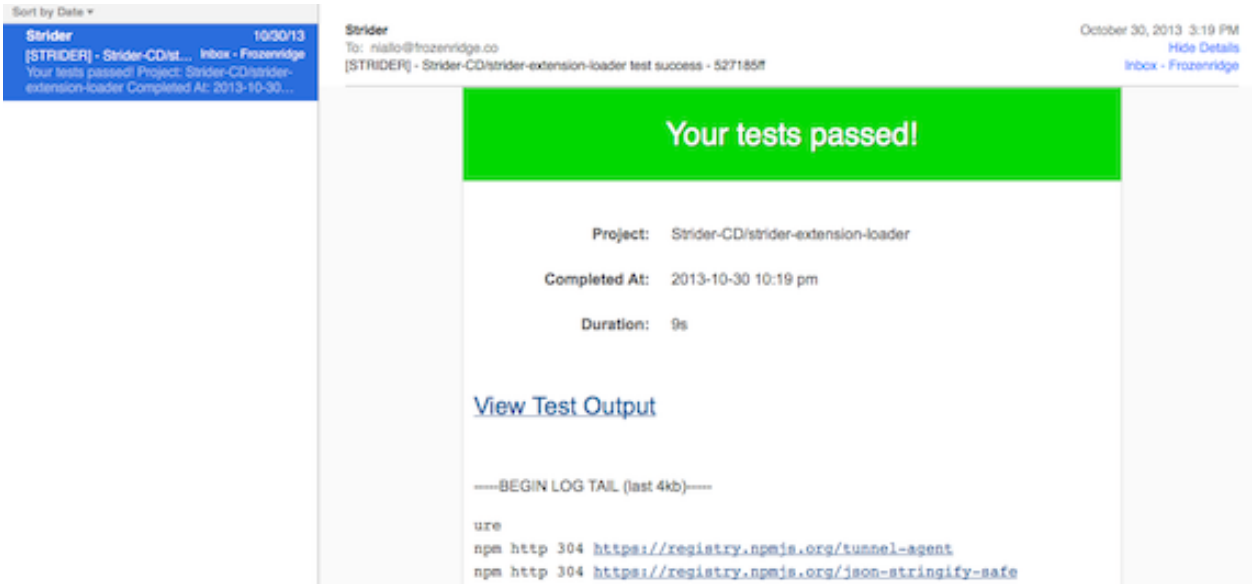
Github, Github Enterprise, BitBucket

Strider integrates with Github, Github Enterprise, BitBucket, GitLab and more. Intuitively add projects for CI and CD with only a few clicks. No messing with scripts or SSH keys - Strider does it for you.



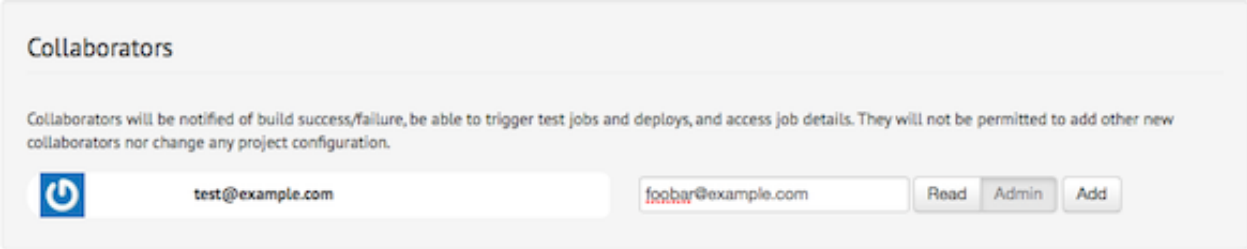
Email And Webhook Notifications

Notification on success and failure. Emails work out of the box to notify humans, Webhooks supported for notifying machines. Want a custom notification method? It's easy to develop a plugin.



Teams and Collaborators

Development teams via "collaborators". Give other users read-only access or full admin rights to your projects.



Public Projects

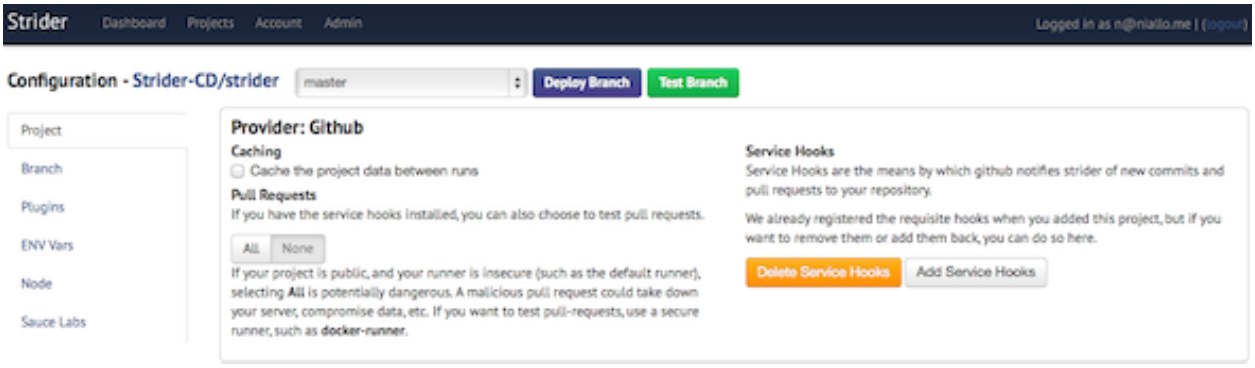
Let the world at large see your project status dashboard! Public projects may be browsed by anonymous users, but not triggered nor configured. Great for Open Source projects. See Strider's public CI dashboard at https://public-ci.stridercd.com.

Branches

Each VCS branch may be configured independently, including different deployment configuration. Use this to create powerful workflows. For example, "master" branch may only deploy to production with a manual trigger while "testing" branches deploy to QA automatically on each successful test run.

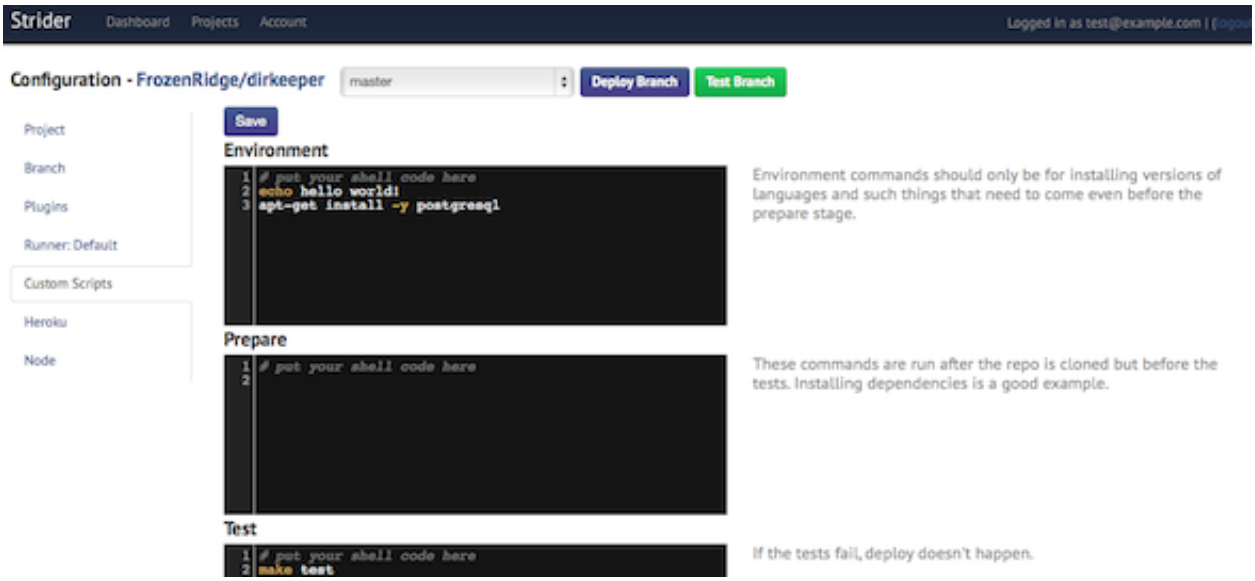
Pull Requests

GitHub pull requests can be automatically tested when they are opened, with test result status sent back to GitHub to mark the PR. For security reasons, this is only enabled for specific users which you whitelist.



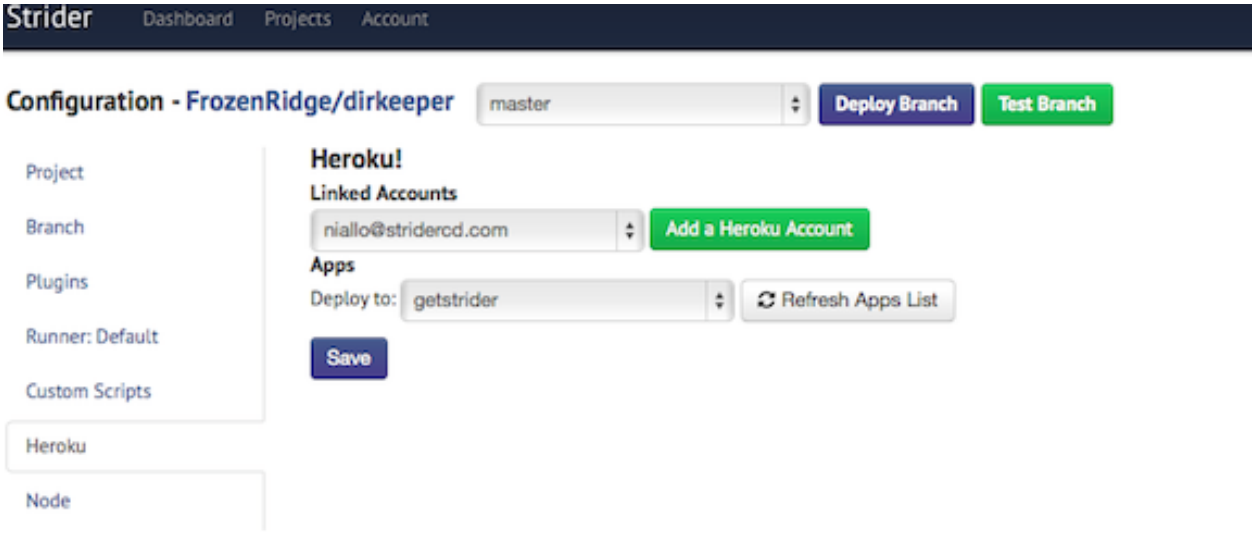
Advanced Shell Scripting

Script each phase of each project from the web or config file. Custom prepare, test and deploy scripts easily enable integration with your specific language, environment and existing automation.



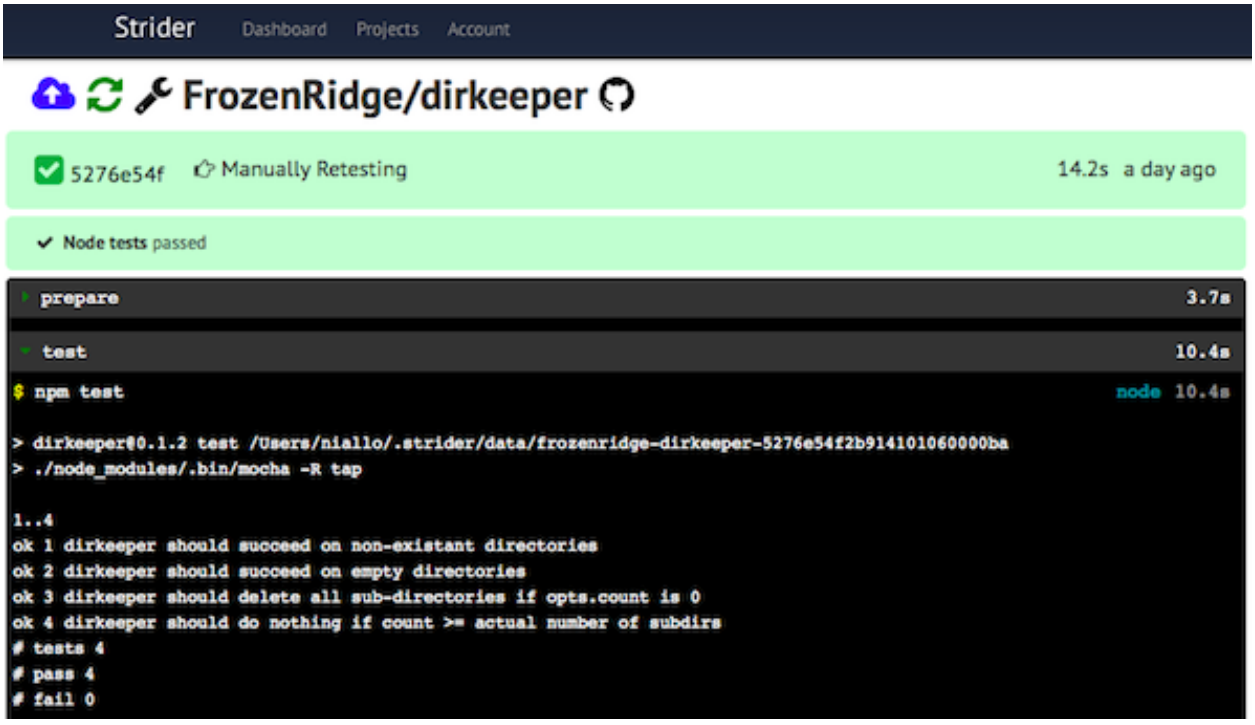
Heroku Support

Deploy directly to Heroku apps with the `strider-heroku` plugin.



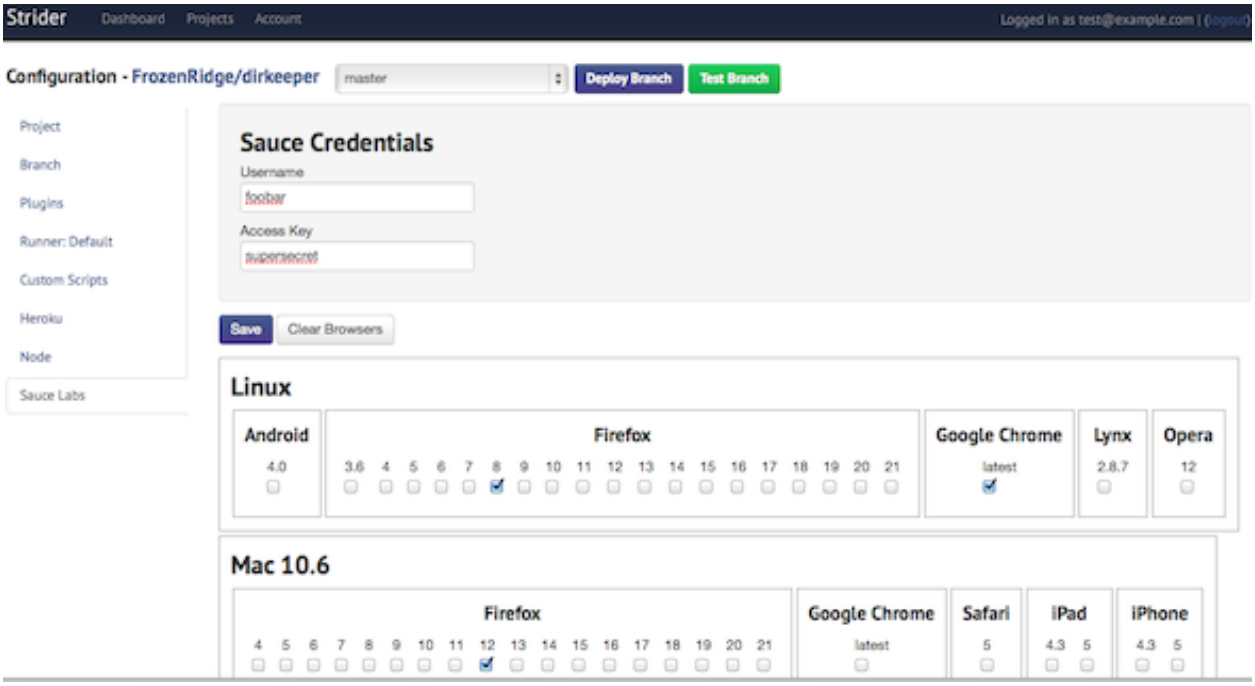
Node.js And Python

Out-of-box support for Node.JS and Python projects. Other languages / environments can be supported with thirdparty plugins or custom scripts.



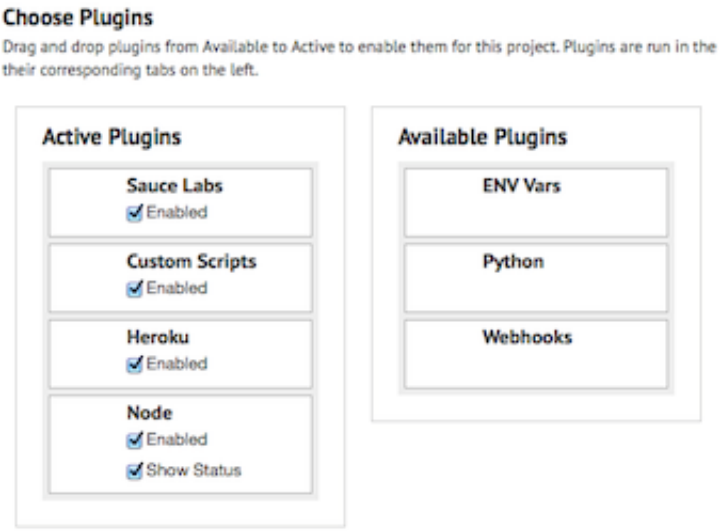
Front-end JavaScript testing with Sauce Labs

Easily configure SauceLabs.com credentials and select os/browser combinations via Strider's web UI. Strider will even manage the Sauce Connect proxy for you - waiting for it to be connected before running your tests.



Extensibility

Strider is designed to be highly extensible. The core is minimal, with most features implemented in plugins (e.g. strider-heroku, strider-github, strider-sauce). In keeping with Node.JS, Strider plugins are simply NPM modules. This makes it easy to support new VCS providers, deployment targets, languages or environments and so on. See the chapter on Extending Strider for more!



Commercial Support

FrozenRidge, LLC provides commercial support including training, customization/integration and cloud-hosting. Email hi@frozenridge.co to learn more.

CHAPTER 2

Installing StriderCD

Dependencies

Strider requires Node 0.10.x or higher and a MongoDB database. You can get node.js for your platform at http://nodejs.org. You can get MongoDB for your platform at http://mongodb.org.

We assume you have a MongoDB server running locally on your machine in the interests of simplicity. Of course, you can use a remote or cloud-hosted MongoDB just as easily.

Latest Stable Version

The latest version of Strider in NPM is always a stable version. Once you have Node.JS on your system, you can install the most recent stable version of Strider system-wide with the following command:

```
npm install -g strider
```

Now you should have a `strider` executable in your `$PATH`.

Latest Development Version

The very latest version of Strider is available on Github. While we strive to keep Strider as stable as possible, this should be considered development or pre-release code.

First clone the repository from Github:

```
git clone https://github.com/Strider-CD/strider.git
```

Next go into the `strider` directory and install strider system-wide:

```
cd strider
npm install -g
```

Now you should have a `strider` executable in your `$PATH`.

Adding Users

Strider isn't much use without any user accounts. You will want to create at least one admin user to manage your instance:

```
strider addUser
```

This command will walk you through creating a new user. If you already know an email and password, you can also pass them directly. For example to create a new user with email `foo@example.com` and password `supersecret` with admin privileges:

```
strider addUser -l foo@example.com -p supersecret -a
```

Cloud-hosted or Remote MongoDB

If you don't want to run a local MongoDB, you can also use a cloud-hosted database, such as a free one provided by MongoLab. Using a cloud-hosted database for Strider can be advantageous because you can easily outsource backups, upgrades and other operations tasks which aren't generally fun.

Additionally, a cloud-hosted MongoDB database should be available from anywhere.

Of course, you might already have your own MongoDB set up in your environment, but just not on your local machine.

Strider can be configured to use a remote MongoDB database with the `DB_URI` environment variable. For example:

```
DB_URI=mongodb://username:supersecret@mongodb.example.com/strider npm start
```

CHAPTER 3

Running Strider

CHAPTER 4

Creating a project

CHAPTER 5

Hacking Strider

Auth

Currently Strider supports two kinds of authentication, cookie and basic. In the near future, Hawk will be added.

Cookie authentication is done by posting to `/login` with a payload of `email` and `password`. Content-type is `www-form-urlencoded`. You then have a session cookie.

Note: say something about expiration...

Basic auth is done by sending the header `"Authorization: basic " + base64(username + ":" + password)`. This must be done for every request that requires authentication.

CORS

Strider supports CORS so that you may call it from a browser or open a websocket to it (Socket.io initiation protocol requires it).

To enable CORS pass a *cors* configuration option when instantiating Strider, which is then passed to the cors npm module.

Example:

```
var strider = require('strider');
var config = {
  cors: {
    origin: 'http://localhost:1337',
    credentials: true,
    headers: [
      'DNT',
      'X-Mx-ReqToken',
      'Keep-Alive',
```

```
      'User-Agent',
      'X-Requested-With',
      'If-Modified-Since',
      'Cache-Control',
      'Content-Type',
      'Accept',
      'Accept-Encoding',
      'Origin',
      'Referer',
      'Pragma',
      'Cookie'
    ],
    methods: [
      'GET',
      'PUT',
      'POST',
      'DELETE',
      'OPTIONS'
    ],
    maxAge: 1728000
  }
};

var app = strider(includePath, config, function(){
  console.log("Strider is running");
});
```

CHAPTER 6

Indices and tables

- search