



JOE JASINSKI - IMAGINARY LANDSCAPE

---

DEPLOYING DJANGO TO  
HEROKU

## OVERVIEW

- ▶ Define 12-Factor App and PaaS
- ▶ Introduce Heroku
- ▶ Heroku Getting Started
- ▶ Integrating with Django
- ▶ Demo
- ▶ Assumes some Django familiarity



THE 12-FACTOR  

---

**APP**

## 12-FACTOR APP ([HTTP://12FACTOR.NET/](http://12factor.net/))

- ▶ 1) Codebase - One codebase tracked in revision control, many deploys
- ▶ 2) Dependencies - Explicitly declare and isolate dependencies
- ▶ 3) Config - Store config in the environment
- ▶ 4) Backing services - Treat backing services as attached resources
- ▶ 5) Build, release, run - Strictly separate build and run stages
- ▶ 6) Processes - Execute the app as one or more stateless processes

## 12-FACTOR APP (CONTINUED)

- ▶ 7) Port binding - Export services via port binding
- ▶ 8) Concurrency - Scale out via the process model
- ▶ 9) Disposability - Maximize robustness with fast startup and graceful shutdown
- ▶ 10) Dev/prod parity - Keep development, staging, and production as similar as possible
- ▶ 11) Logs - Treat logs as event streams
- ▶ 12) Admin processes - Run admin/management tasks as one-off processes



PLATFORM AS A  
SERVICE

---

**PAAS**

## PLATFORM AS A SERVICE

-  ▶ Run application without managing infrastructure
-  ▶ Reduced complexity
-  ▶ Scales well
-  ▶ Less control over infrastructure
- ▶ Examples:
  - ▶ Heroku, Amazon Elastic Beanstalk, Google App Engine, OpenShift



---

HEROKU

## ABOUT HEROKU

- ▶ Platform as a Service (PaaS)
- ▶ 12-Factor Design
- ▶ Deploy via Git
- ▶ Easy to Use!
- ▶ Supported Languages
  - ▶ Ruby
  - ▶ Python
  - ▶ Nodes
  - ▶ Go
  - ▶ PHP
  - ▶ Java
  - ▶ Scala
  - ▶ Clojure

## DYNOS

- ▶ 1 Dyno = 1 running process
- ▶ Executes your code
- ▶ Like a server, but completely disposable
- ▶ Price based on num of Dynos used
- ▶ Single Dyno is free

# HEROKU ADDONS

Data Stores

Choose where to store your data.

 Graph Story Enterprise Neo4j Graph Databases as a Service	 Redis Cloud Enterprise-Class Redis for Developers	 AT&T M2X AT&T's time-series data storage service for the Internet of Things	 ClearDB MySQL The high speed database for your MySQL powered
 Instaclustr Hosted and managed Apache Cassandra NoSQL databases	 RedisGreen Production-quality Redis servers with superior support and	 openredis Dependable Redis Hosting.	 GrapheneDB Neo4j Graph Database as a Service
			

[HTTPS://ELEMENTS.HEROKU.COM/ADDONS](https://elements.heroku.com/addons)



GETTING  

---

**STARTED**

# SIGN UP

► Register

► Confirm Email

► Set Password

► All ready!



Sign up for free and experience Heroku today

Free account

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

Your app platform

A platform for apps, with app management & instant scaling, for development and production.

Deploy now

Go from code to running app in minutes. Deploy, scale, and deliver your app to the world.

First name

Last name

Email

Company name

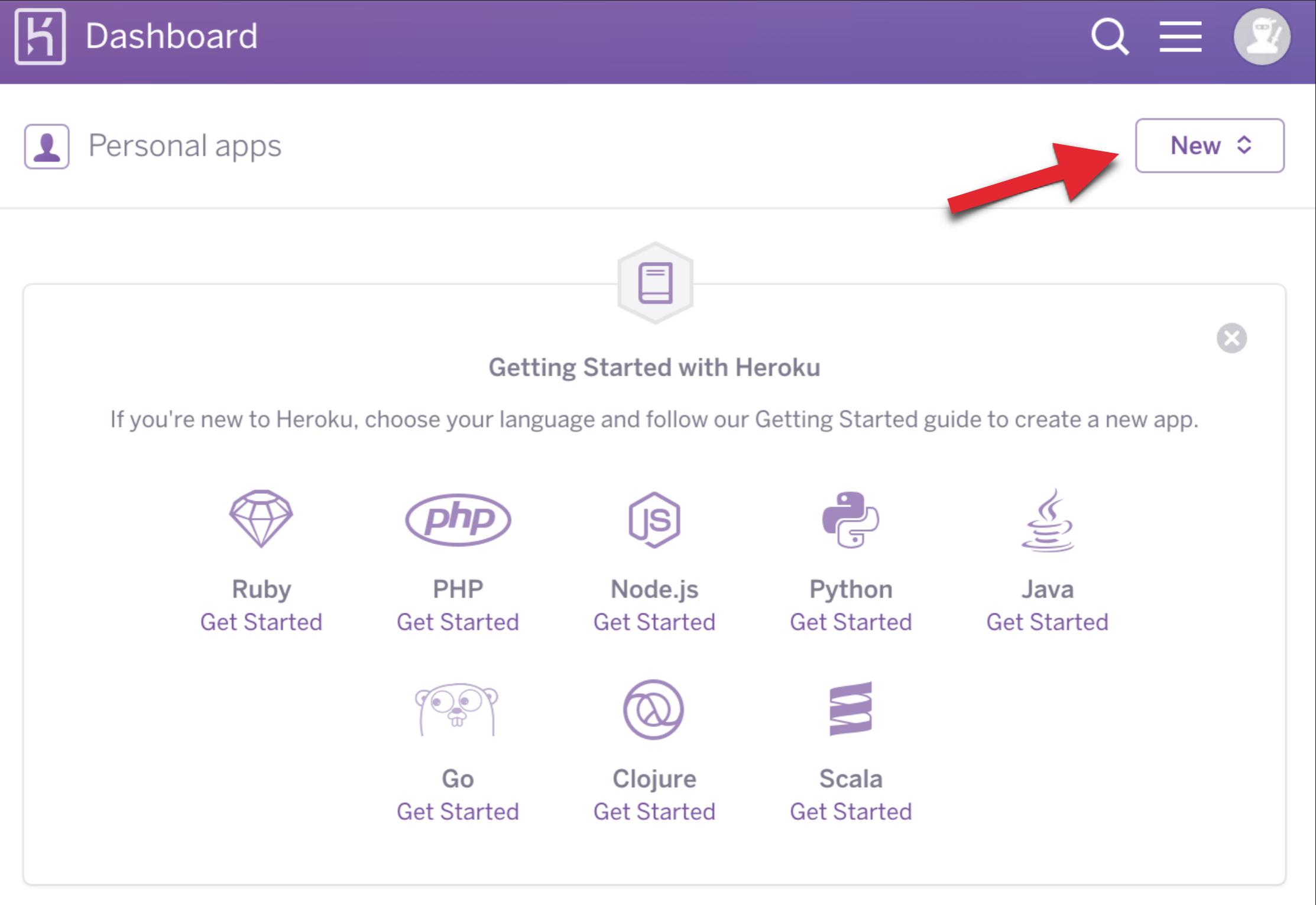
Pick your primary development language

Select a Language ▾

**Create Free Account**

Signing up signifies that you have read and agree to  
the [Terms of Service](#) and [Privacy Policy](#).

# CREATE NEW APP



The screenshot shows the Heroku Dashboard interface. At the top left is the Heroku logo and "Dashboard" text. On the right side of the header are a search icon, a menu icon, and a user profile icon. Below the header, there's a section titled "Personal apps" with a user icon. To the right of this is a "New" button with a dropdown arrow, which is highlighted by a large red arrow pointing towards it from the bottom right. A modal window titled "Getting Started with Heroku" is open in the center. It contains the text: "If you're new to Heroku, choose your language and follow our Getting Started guide to create a new app." Below this, there are five language icons with their names and "Get Started" links: Ruby (diamond icon), PHP (PHP icon), Node.js (JS icon), Python (Python icon), and Java (coffee cup icon). At the bottom of the modal, there are three more language icons with "Get Started" links: Go (gorilla icon), Clojure (clojure icon), and Scala (Scala icon).

Dashboard

Personal apps

New

Getting Started with Heroku

If you're new to Heroku, choose your language and follow our Getting Started guide to create a new app.

Ruby  
Get Started

PHP  
Get Started

Node.js  
Get Started

Python  
Get Started

Java  
Get Started

Go  
Get Started

Clojure  
Get Started

Scala  
Get Started

# CREATE NEW APP

The screenshot shows the Heroku Dashboard interface for creating a new application. At the top, there's a purple header bar with the Heroku logo, the word "Dashboard", and user profile icons. Below the header, the main area has a white background with a light gray horizontal line separating sections.

**New App**

**App Name (optional)**  
Leave blank and we'll choose one for you.

A text input field contains the value "chipy-dev2". A green rectangular box surrounds this input field, and a red oval highlights the entire input area. Below the input field, the text "chipy-dev2 is available" is displayed in green.

**Runtime Selection**  
Your app can run in your choice of region in the Common Runtime.

A dropdown menu is open, showing "United States" as the selected option.

**Create App**

# CREATE NEW APP

Dashboard

Personal apps  
chipy-dev2 ★

Open app More

Overview Resources Deploy Metrics Activity Access Settings

Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. [Learn more.](#)

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. [Learn more.](#)

New Pipeline... Add to a Pipeline

Deployment method

Heroku Git Use Heroku Toolkit

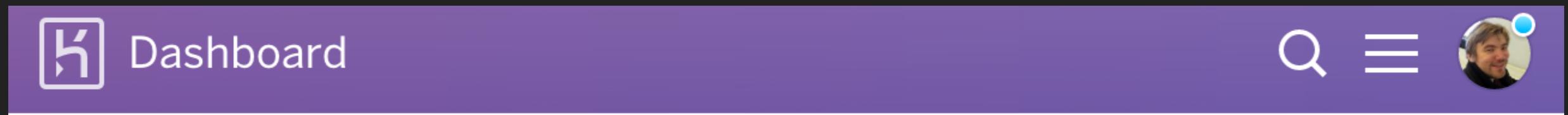
GitHub Connect to GitHub

Dropbox Connect to Dropbox

A screenshot of the Heroku Dashboard for the app 'chipy-dev2'. The dashboard has a purple header with the Heroku logo, a search bar, and a user profile icon. Below the header, there are tabs for Overview, Resources, Deploy (which is selected), Metrics, Activity, Access, and Settings. On the left, there's a sidebar with a 'Personal apps' section containing 'chipy-dev2' with a star icon. To the right of the sidebar, there are two main sections: 'Add this app to a pipeline' and 'Add this app to a stage in a pipeline to enable additional features'. The 'Add this app to a pipeline' section includes a description, a 'New Pipeline...' button, and an 'Add to a Pipeline' button. The 'Deployment method' section at the bottom shows three options: 'Heroku Git' (selected), 'GitHub' (with 'Connect to GitHub'), and 'Dropbox' (with 'Connect to Dropbox'). A large red arrow points from the 'Deployment method' heading down to the 'Heroku Git' button.

- ▶ ON you desktop:
- ▶ Download and Install Heroku Toolbelt
  - ▶ <https://devcenter.heroku.com/articles/getting-started-with-python#set-up>
- ▶ heroku login

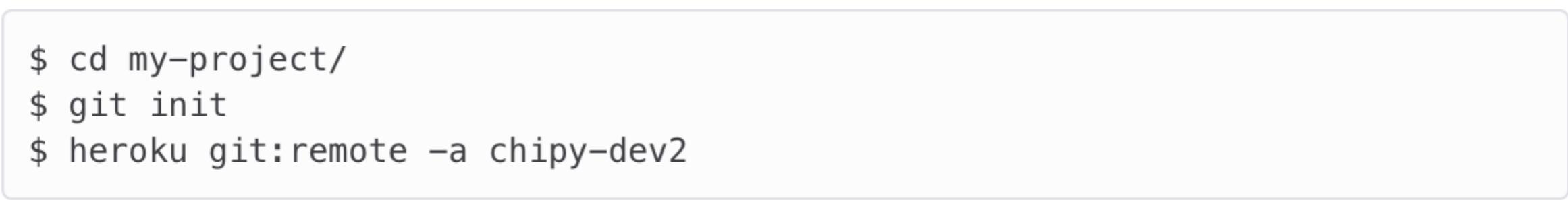
# BASIC DEPLOYMENT WORKFLOW



\$ heroku login

Create a new Git repository

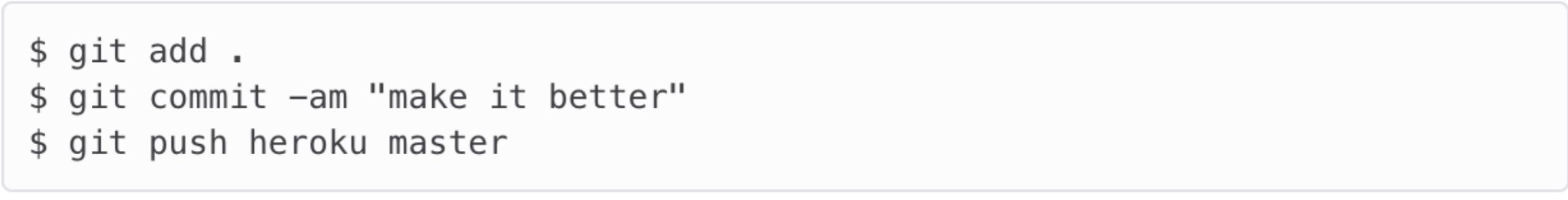
Initialize a git repository in a new or existing directory



```
$ cd my-project/
$ git init
$ heroku git:remote -a chipy-dev2
```

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.



```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

<http://chipy-dev2.herokuapp.com/>



HEROKU WITH

---

DJANGO

# A FEW NEEDED FILES (AT PROJECT ROOT)

- ▶ Procfile

- ▶ `web: gunicorn deploy.wsgi --pythonpath chipy_org --log-file -`
- ▶ <https://goo.gl/5rlhE9>

- ▶ requirements.txt

- ▶ specifies exact python dependencies

- ▶ i.e. `Django==1.9.6`

- ▶ <https://goo.gl/XTkJPx>

- ▶ .gitignore

# APP CONFIGURATION

# CONFIGURATION VARIABLES

- ▶ Contain deploy-specific information
- ▶ Configuration is set via UNIX environment variables
- ▶ Environment variables can be set via command line:
  - ▶ i.e. `heroku config:set DEBUG=True`
  - ▶ Or they can be set via Heroku Admin....

# CONFIGURATION VARIABLES

The screenshot shows the Heroku dashboard for the 'chipy-dev' app. A red arrow points from the 'Settings' tab in the navigation bar to the 'Environment Variables' section.

**Dashboard** Personal apps chipy-dev ★

Overview Resources Deploy Metrics Activity Access Settings

Name chipy-dev

Config Variables

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

ADMINS

ALLOWED\_HOSTS

AWS\_ACCESS\_KEY\_ID

AWS\_SECRET\_ACCESS\_KEY

**Environment Variables**

Variable	Value	Action	
ADMINS	[REDACTED]		
ALLOWED_HOSTS	chipy.org, www.chipy.org, chipy-dev		
AWS_ACCESS_KEY_ID	[REDACTED]		
AWS_SECRET_ACCESS_KEY	[REDACTED]		

# CONFIG VARS IN SETTINGS.PY

<https://goo.gl/GTRnbd>

```
9
10 def env_var(key, default=None):
11     """Retrieves env vars and makes Python boolean replacements"""
12     val = os.environ.get(key, default)
13     if val in ('True', 'true'):
14         val = True
15     elif val in ('False', 'false'):
16         val = False
17     return val
18
19
20 def env_list(key, defaults=[], delimiter=','):
21     val_list = defaults
22     val = os.environ.get(key, None)
23     if val:
24         val_list = val.split(delimiter)
25     return val_list
26
27
28 PROJECT_ROOT = os.path.abspath(os.path.dirname(__file__))
29
30 DEBUG = env_var('DEBUG', False)
31
32 ALLOWED_HOSTS = ['chipy.org', 'www.chipy.org', 'chipy.herokuapp.com', 'chipy'
33                 'localhost:8000', 'www.localhost:8000', 'www.localhost']
34
35 ALLOWED_HOSTS = env_list("ALLOWED_HOSTS", ALLOWED_HOSTS)
36
37 GITHUB_APP_ID = env_var('GITHUB_APP_ID')
38 GITHUB_API_SECRET = env_var('GITHUB_API_SECRET')
```

JOE JASINSKI

---

# DATABASE SETUP

# CREATING A DATABASE

The screenshot shows a software interface with a purple header bar. On the left, there's a sidebar titled "Personal apps" containing a list of applications: "chipy", "chipy-dev", "chipy-voter", "joe-erin", "joejasinski-test", and "nameless-tor-2565". To the right of the sidebar is a search bar, a menu icon, and a user profile picture. A dropdown menu is open, listing several options: "Dashboard", "Databases", "Dataclips", "Elements", "Documentation", and "Support".

- Dashboard
- Databases
- Dataclips
- Elements
- Documentation
- Support

# DATABASE SETTINGS

The screenshot shows the Heroku Postgres Overview page for a database named "heroku-postgres-14932c09 :: red". The connection settings are highlighted with a red rounded rectangle:

Setting	Value
Host	ec2-23-21-66-88.compute-1.amazonaws.com
Database	d7fet2pihh13c4
User	gipjwptopeiuou
Port	5432
Password	<a href="#">Show</a>
Psycopg	heroku pg:psycopg --app heroku-postgres-14932c09 HEROKU_POSTGRESQL_RED
URL	<a href="#">Hide</a> postgres://gipjwptopeiuou:L-Ey_wBr9jtOUQgDFU-ywyFeFI@ec2-23-21-66-88.compute-1.amazonaws.com:5432/d7fet2pihh13c4

Below the connection settings, a note states: "There are a variety of ways to connect to a Heroku Postgres database. Find out how to do so via [psycopg](#), [Java](#), [Ruby](#), [Python](#), or [Node.js](#)".

## DJ DATABASE URL

- ▶ <https://pypi.python.org/pypi/dj-database-url>
- ▶ pip install dj-database-url # ... and add to requirements.txt
- ▶ Add the following to settings.py

```
52  
53 # dj_database_url will pull from the DATABASE_URL environment variable  
54 DATABASES = {  
55     'default': dj_database_url.config(default='postgres://localhost:5432/chipy_org')}  
56
```

- ▶ Set the DATABASE\_URL environment variable

## DJ DATABASE URL (CONT)

- ▶ heroku config:set DATABASE\_URL=\  
postgres://USER:PASS@HOST:5432/NAME

Engine	Django Backend	URL
PostgreSQL	django.db.backends.postgresql_psycopg2	postgres://USER:PASSWORD@HOST:PORT/NAME
PostGIS	django.contrib.gis.db.backends.postgis	postgis://USER:PASSWORD@HOST:PORT/NAME
MySQL	django.db.backends.mysql	mysql://USER:PASSWORD@HOST:PORT/NAME
MySQL (GIS)	django.contrib.gis.db.backends.mysql	mysqlgis://USER:PASSWORD@HOST:PORT/NAME
SQLite	django.db.backends.sqlite3	<a href="#">sqlite:///PATH</a>
Oracle	django.db.backends.oracle	oracle://USER:PASSWORD@HOST:PORT/NAME
Oracle (GIS)	django.contrib.gis.db.backends.oracle	oraclegis://USER:PASSWORD@HOST:PORT/NAME

STATICFILE  
AND  
UPLOADED MEDIA HOSTING

## STATIC FILE HOSTING

- ▶ pip install whitenoise # ... and add to requirements.txt
- ▶ edit settings.py

```
MIDDLEWARE_CLASSES = [  
    # 'django.middleware.security.SecurityMiddleware',  
  
    'whitenoise.middleware.WhiteNoiseMiddleware',  
]  
  
STATIC_ROOT = os.path.join(PROJECT_ROOT, 'staticfiles')  
  
STATIC_URL = '/static/'  
  
STATICFILES_STORAGE = \  
    'whitenoise.django.GzipManifestStaticFilesStorage'
```

## DJANGO STORAGES

- ▶ S3, CloudFiles, Libcloud, sFTP, Azure Storage, etc.
- ▶ Django Storages / Django Storages Redux
  - ▶ <https://github.com/jschneier/django-storages>
  - ▶ <https://pypi.python.org/pypi/django-storages-redux>
- ▶ pip install django-storages-redux # ... and add to req
- ▶ pip install boto # ... and add to requirements.txt

# DJANGO STORAGES - CONFIGURE WITH S3

► <https://goo.gl/sQcLiK>

```
192 INSTALLED_APPS = [  
...  
221     'storages',  
222     'tinymce',  
223     "sorl.thumbnail",  
224 ]
```

```
91     AWS_ACCESS_KEY_ID = os.environ.get('AWS_ACCESS_KEY_ID')  
92     AWS_SECRET_ACCESS_KEY = os.environ.get('AWS_SECRET_ACCESS_KEY')  
93     AWS_STORAGE_BUCKET_NAME = os.environ.get('AWS_STORAGE_BUCKET_NAME')  
94     AWS_QUERYSTRING_AUTH = False  
95     AWS_HEADERS = {  
96         'Cache-Control': 'max-age=86400',  
97     }  
98     DEFAULT_FILE_STORAGE = 'storages.backends.s3boto.S3BotoStorage'
```

JOE JASINSKI

---

# SENDING EMAIL

## SENDGRID ADDON

- ▶ free Tier but needs credit card
- ▶ heroku addons:create sendgrid:starter
- ▶ edit settings.py

```
248
249 EMAIL_BACKEND = env_var('EMAIL_BACKEND', 'django.core.mail.backends.smtp.EmailBackend')
250 EMAIL_HOST = env_var('EMAIL_HOST', 'smtp.sendgrid.net')
251 EMAIL_HOST_USER = env_var('EMAIL_HOST_USER', env_var('SENDGRID_USERNAME', None))
252 EMAIL_HOST_PASSWORD = env_var('EMAIL_HOST_PASSWORD', env_var('SENDGRID_PASSWORD', None))
253 EMAIL_PORT = int(env_var('EMAIL_PORT', 587))
254 EMAIL_USE_TLS = env_var('EMAIL_USE_TLS', True)
```

- ▶ Heroku sets USERNAME/PASSWORD environment variables:
  - ▶ heroku config:get SENDGRID\_USERNAME --app chipy-dev
  - ▶ heroku config:get SENDGRID\_PASSWORD --app chipy-dev

## SENDGRID ADDON (CONT)

- ▶ Then send email within the app like normal:

```
from django.core.mail import send_mail  
  
send_mail(  
    "My Subject",  
    "My Content",  
    'noreply@chipy-dev.herokuapp.com',  
    ['example@gmail.com'])
```

# COMMON HEROKU COMMANDS

# COMMON COMMANDS

```
# Deploy changes to master  
git push heroku master
```

```
# Deploy feature branch  
git push heroku feature/mybranch:master
```

```
# Collectstatic  
heroku run python manage.py collectstatic --  
noinput
```

```
# Set migrate the database  
heroku run python manage.py migrate
```

```
# Set environment variable on Heroku  
heroku config:set DEBUG=True
```

```
# Show Config variables  
heroku config
```

```
# Get Dyno Logs  
heroku logs -tail
```

```
# Get to Django Shell  
heroku run manage.py shell
```

```
# Specify command for a specific app  
heroku -app chipy-dev ...
```

## NEXT GO HERE...

- ▶ Integrating Django with Heroku
  - ▶ <https://devcenter.heroku.com/articles/django-app-configuration>
- ▶ Working Open Source Example ([chipy.org](http://chipy.org))
  - ▶ <https://github.com/chicagopython/chipy.org>

HEROKU CREATE CHIPY-DEV2

GIT REMOTE ADD HEROKU2 HTTPS://GIT.HEROKU.COM/CHIPIY-DEV2.GIT

---

HEROKU CONFIG:SET DEBUG=TRUE --APP CHIPY-DEV2

HEROKU CONFIG:SET ALLOWED\_HOSTS="CHIPIY.ORG,WWW.CHIPIY.ORG" --APP CHIPY-DEV2

HEROKU CONFIG:SET GITHUB\_APP\_ID=YOURIDHERE --APP CHIPY-DEV2

HEROKU CONFIG:SET GITHUB\_API\_SECRET=SUPERSECRETKEYHERE --APP CHIPY-DEV2

HEROKU CONFIG:SET SECRET\_KEY=SOMESECRETKEYFORDJANGOGOESHHERE --APP CHIPY-DEV2

HEROKU CONFIG:SET ADMINS=ADMIN@EXAMPLE.COM --APP CHIPY-DEV2

HEROKU CONFIG:SET ENVELOPE\_EMAIL\_RECIPIENTS=ADMIN@EXAMPLE.COM --APP CHIPY-DEV2

HEROKU CONFIG:SET NORECAPTCHA\_SITE\_KEY=YOUR\_RECAPTCHA\_PUBLIC\_KEY --APP CHIPY-DEV2

HEROKU CONFIG:SET NORECAPTCHA\_SECRET\_KEY=YOUR\_RECAPTCHA\_PRIVATE\_KEY --APP CHIPY-DEV2

HEROKU ADDONS:CREATE HEROKU-POSTGRESQL:HOBBY-DEV --APP CHIPY-DEV2

GIT PUSH HEROKU2 MASTER

HEROKU RUN PYTHON MANAGE.PY MIGRATE AUTH 0004 --APP CHIPY-DEV2

HEROKU RUN PYTHON MANAGE.PY MIGRATE --APP CHIPY-DEV2

HEROKU APPS:DESTROY --APP CHIPY-DEV2

HEROKU CREATE CHIPY-DEV2