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/)

- 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

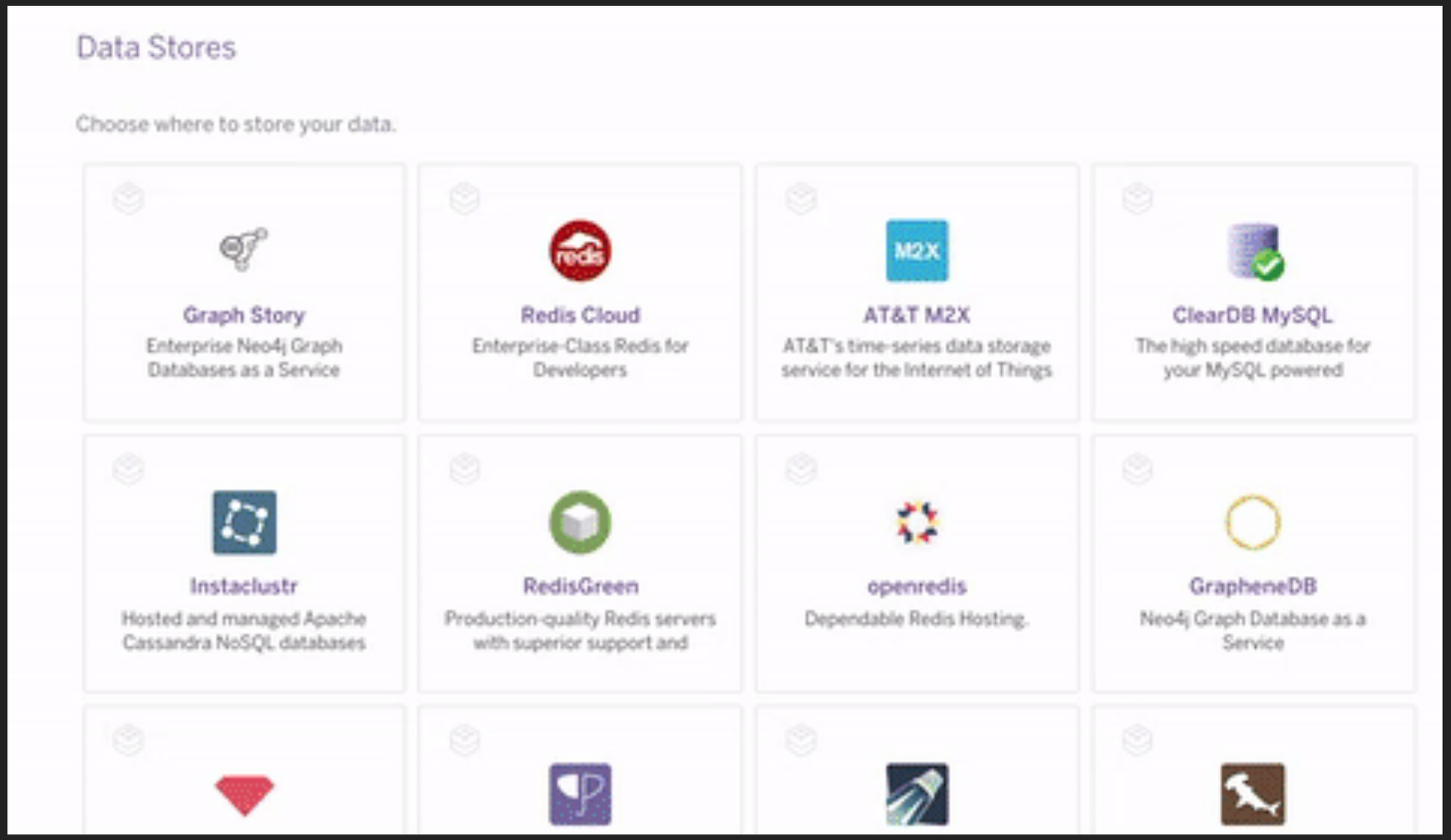
- Platform as a Service (PaaS)
- 12-Factor Design
- Deploy via Git
- Easy to Use!

- 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



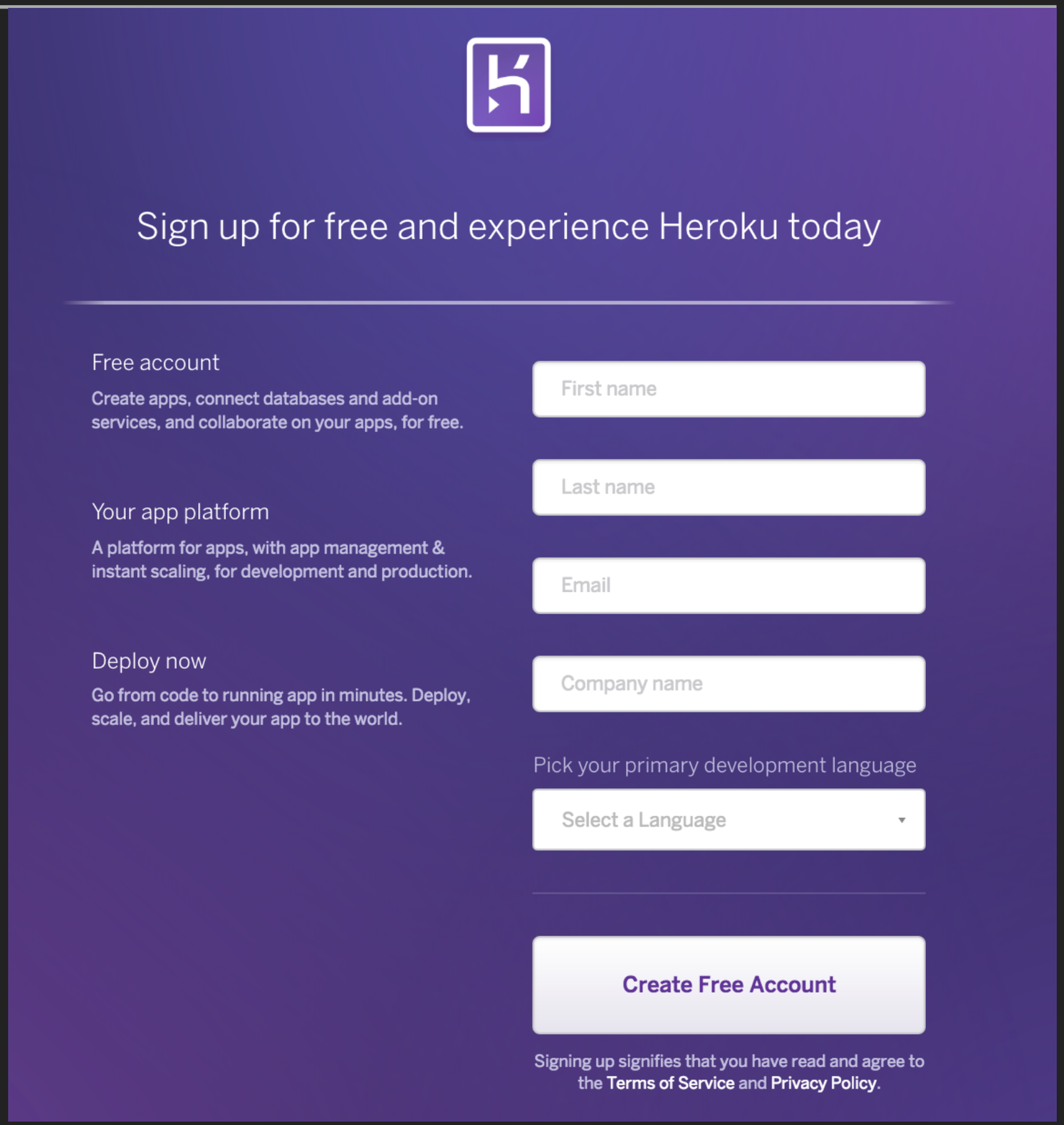
HTTPS://ELEMENTS.HEROKU.COM/ADDONS

GETTING

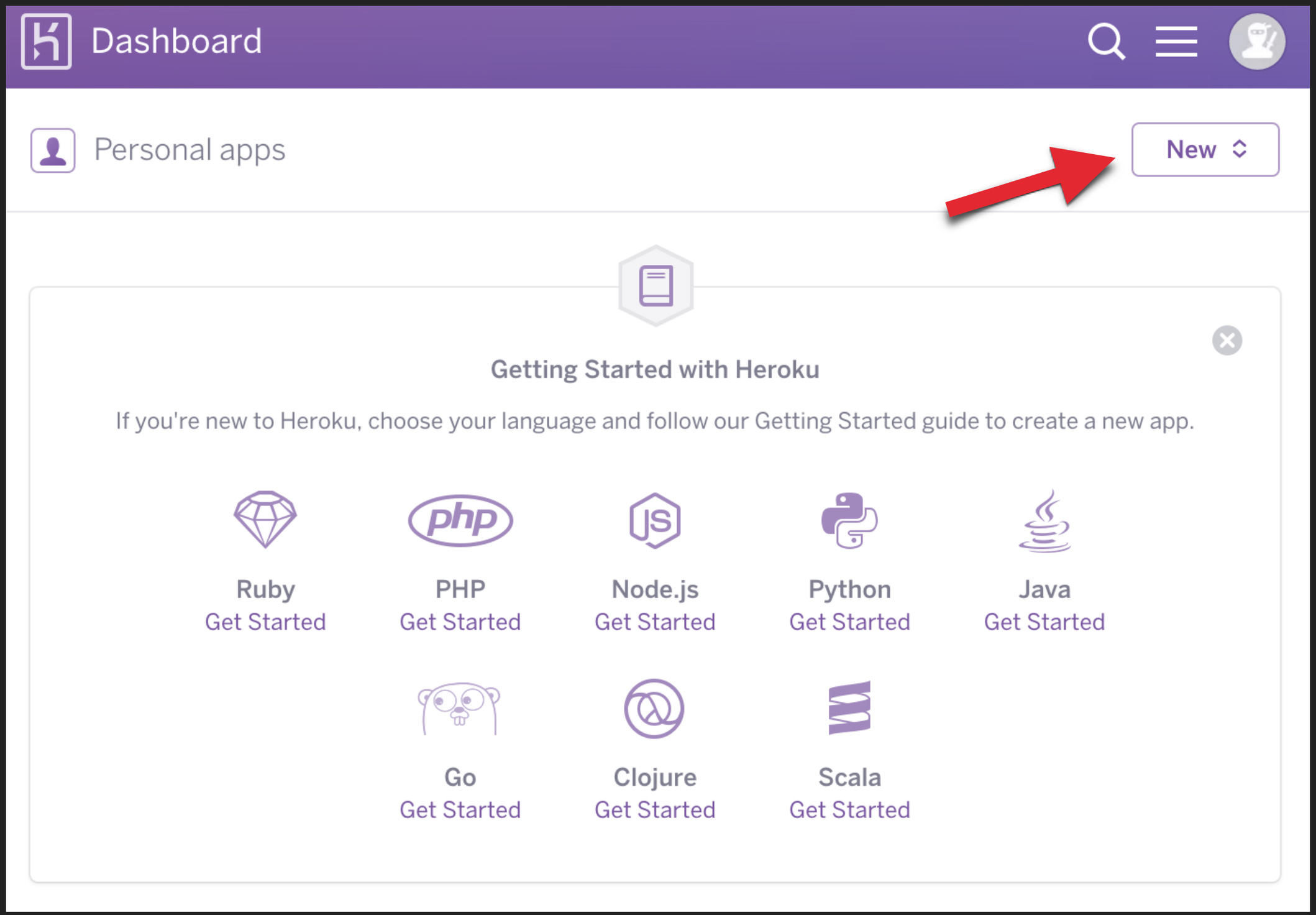
STARTED

SIGN UP

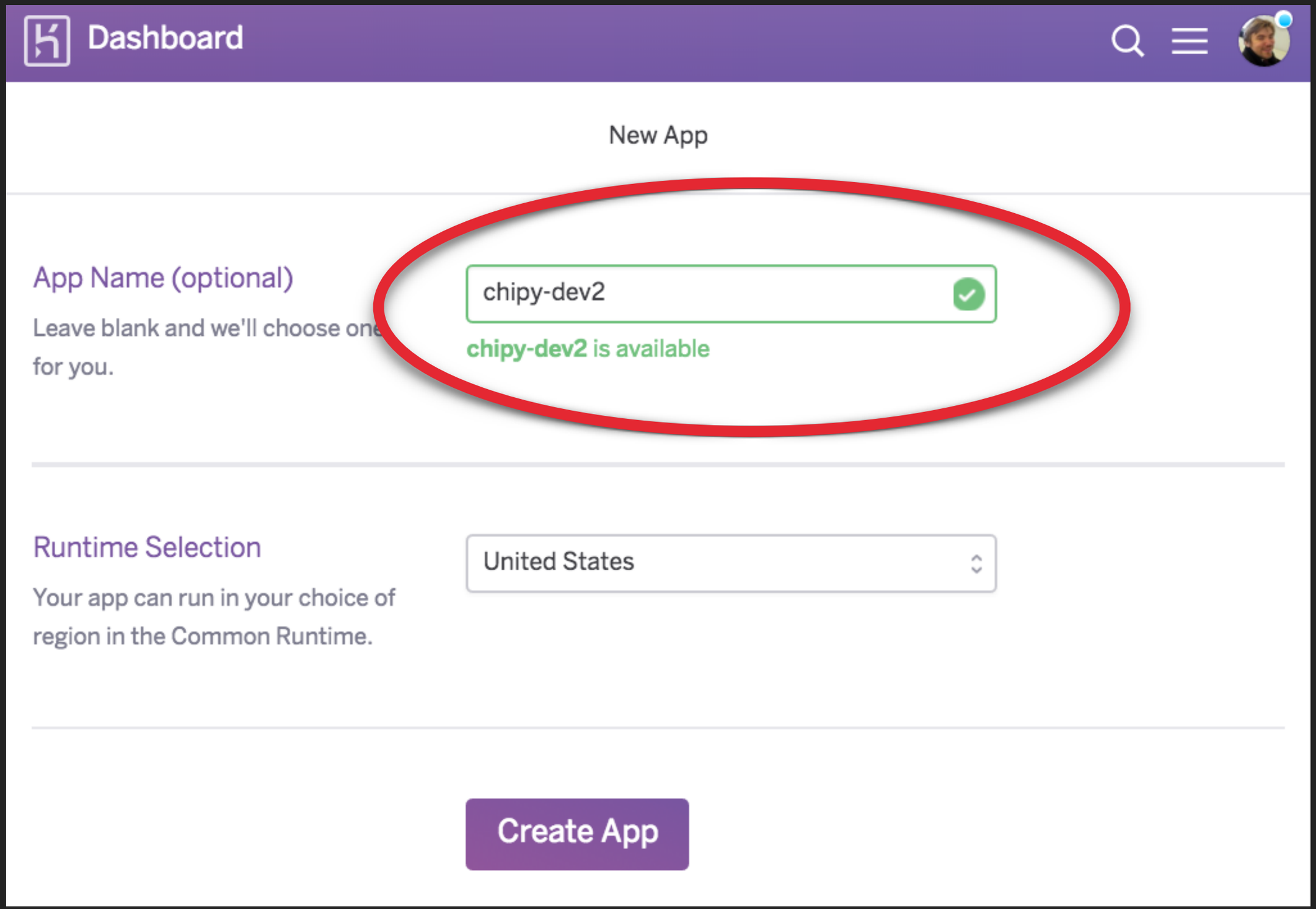
- Register
- Confirm Email
- Set Password
- All ready!



CREATE NEW APP



CREATE NEW APP



JOE JASINSKI

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 Toolbelt

GitHub
Connect to GitHub

Dropbox
Connect to Dropbox

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

JOE JASINSKI

BASIC DEPLOYMENT WORKFLOW

Dashboard

```
$ 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

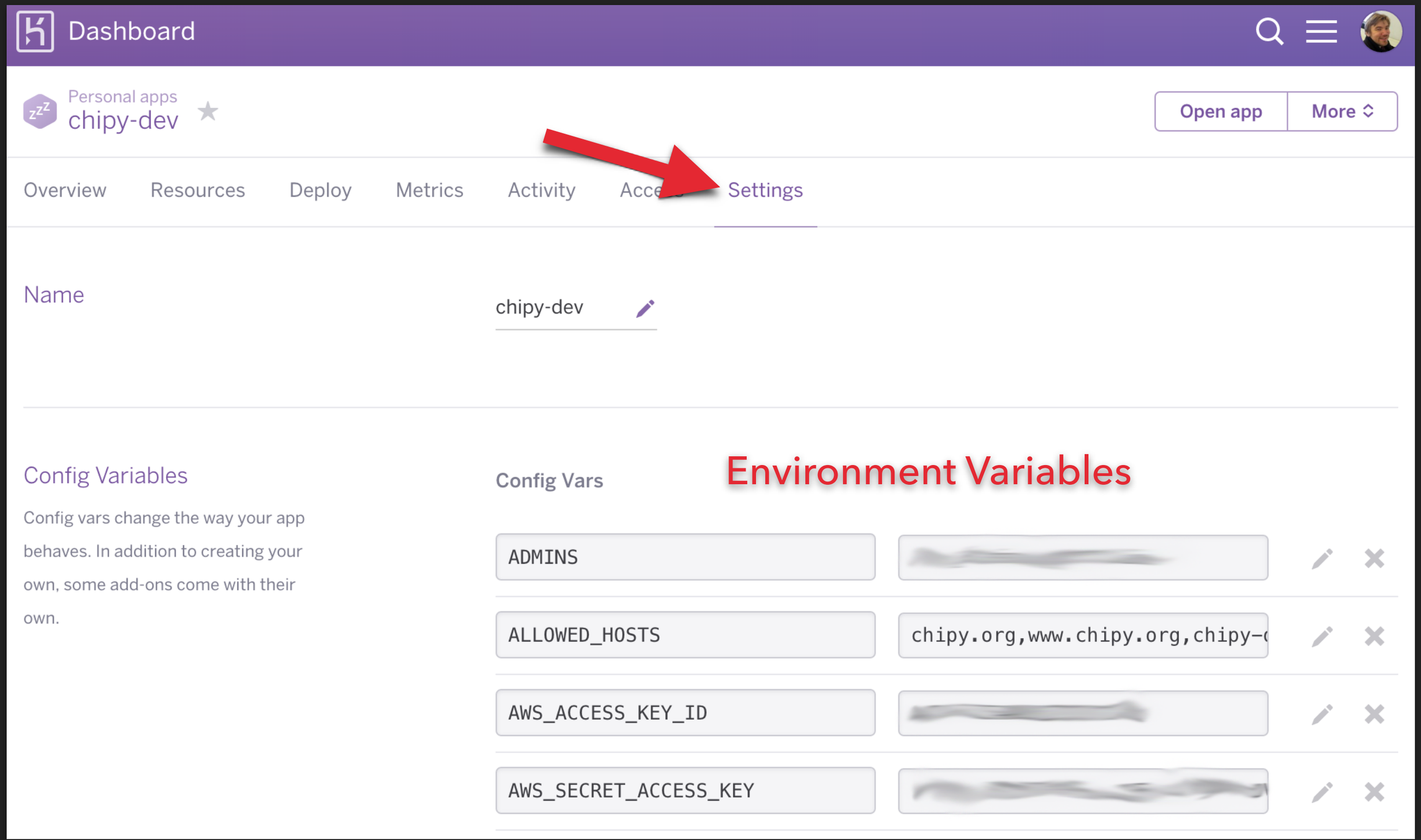
APP CONFIGURATION

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….

JOE JASINSKI

CONFIGURATION VARIABLES



JOE JASINSKI

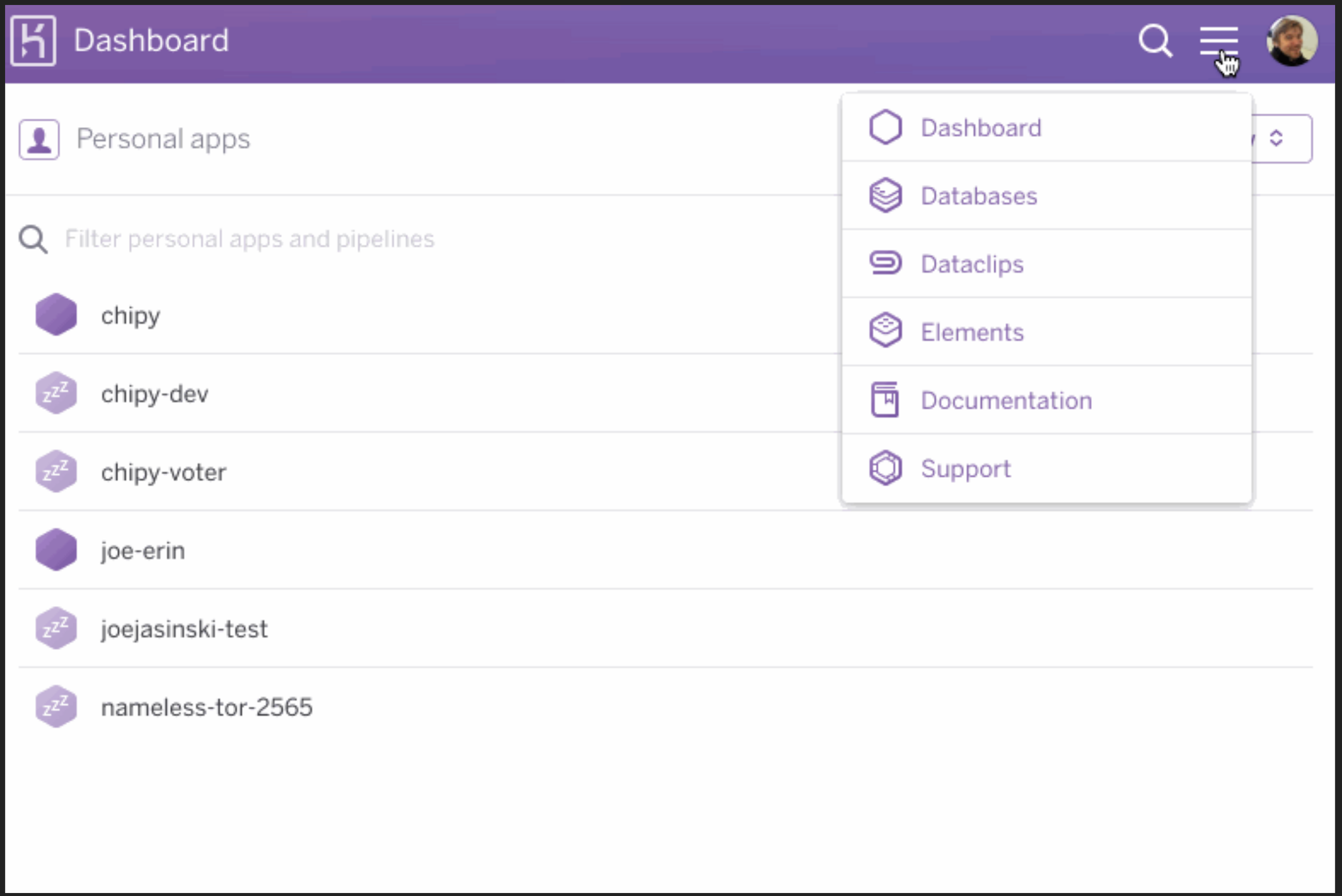
CONFIG VARS IN SETTINGS.PY

https://goo.gl/GTRnbd

```python
def env_var(key, default=None):
    """Retrieves env vars and makes Python boolean replacements"""
    val = os.environ.get(key, default)
    if val in ('True', 'true'):
        val = True
    elif val in ('False', 'false'):
        val = False
    return val


def env_list(key, defaults=[], delimiter=','):
    val_list = defaults
    val = os.environ.get(key, None)
    if val:
        val_list = val.split(delimiter)
    return val_list


PROJECT_ROOT = os.path.abspath(os.path.dirname(__file__))

DEBUG = env_var('DEBUG', False)

ALLOWED_HOSTS = ['chipy.org', 'www.chipy.org', 'chipy.herokuapp.com', 'ch
                 'localhost:8000', 'www.localhost:8000', 'www.localhost']

ALLOWED_HOSTS = env_list("ALLOWED_HOSTS", ALLOWED_HOSTS)

GITHUB_APP_ID = env_var('GITHUB_APP_ID')
GITHUB_API_SECRET = env_var('GITHUB_API_SECRET')
```

DATABASE SETUP

CREATING A DATABASE



DATABASE SETTINGS

Heroku Postgres

heroku-postgres-14932c09 :: red

Overview PG Backups

Connection Settings

Host	ec2-23-21-66-88.compute-1.amazonaws.com
Database	d7fet2pihh13c4
User	gipjwptopeiuou
Port	5432
Password	Show
Psql	heroku pg:psql --app heroku-postgres-14932c09 HEROKU_POSTGRESQL_RED
URL	Hide postgres://gipjwptopeiuou:L-Ey_wBr9jt0UQgDFU-ywyFeFI@ec2-23-21-66-88.compute-1.amazonaws.com:5432/d7fet2pihh13c4

There are a variety of ways to connect to a Heroku Postgres database. Find out how to do so via psql, 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`	`sqlite:///PATH`
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'
```

SENDING EMAIL

SENDGRID ADDON

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

```
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)
  - https://github.com/chicagopython/chipy.org

HEROKU CREATE CHIPY-DEV2

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

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

HEROKU CONFIG:SET ALLOWED_HOSTS="CHIPY.ORG,WWW.CHIPY.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=SOMESECRETKEYFORDJANGOGOESHERE --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