



DEPLOYING TO THE WEB

WE AREN'T LOCALHOST DEVELOPERS, ARE WE?

- ▶ We will try our hands at getting something to the web.
- ▶ We don't move on until *everyone* is on the same page.
- ▶ This is a potential headache. Bear with me.
- ▶ First, a segue on branching in git.

BRANCHING IN GIT

GO TO THE BLOGR APP

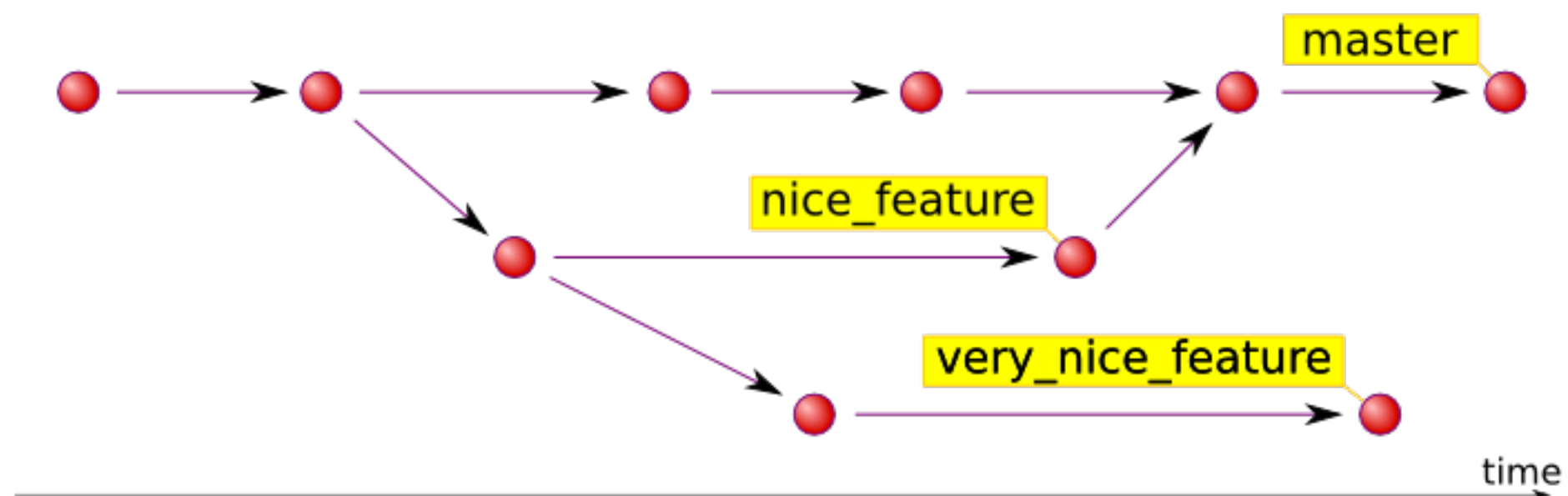
- ▶ If you haven't finished this in class, you can find it on github
- ▶ `mv` this out of the class directory if you do have it in there
- ▶ If you have it inside another app it will become a submodule, which is pretty annoying IMHO.
- ▶ In the blogr folder run `git init`

GIT ENABLES US TO FAIL SAFELY

- ▶ Perhaps at some point you have a nice clean working app, like now.
- ▶ Then you try some feature and you completely wreck things
- ▶ Git lets you get back to normal.

BEFORE YOU TRY THAT NEW FEATURE, MAKE A BRANCH

- ▶ Master should be perfect at all times. Master should be ready to deploy to the web.
- ▶ Before you do something new, create a branch



- ▶ `git status` (should be master)
- ▶ `git add .`
- ▶ `git commit -m 'initial commit'` (I recommend doing this right after rails new and then making additional commits later)
- ▶ `git status`

- ▶ `git checkout -b new-site-title`
- ▶ `git status`
- ▶ `git branch`

WHAT WE JUST DID

- ▶ We created this alternate universe for your code.
- ▶ Feel free to mess around. Whatever happens, you can go back to master and everything is just as you left it.
- ▶ When this branch is up to par, then you can merge it in.
- ▶ Let's change the title that appears in the browser tab

DEPLOYING

HEROKU

- ▶ Cloud platform as a service
- ▶ Provides containerized (dyno) servers to developers.
- ▶ Owned by Salesforce
- ▶ Employs Matsumoto Yukihiro
- ▶ Allows free app deployment, and server management.
- ▶ Up to 10,000 rows of DB free
- ▶ Solves most devops problems for you



GET AN ACCOUNT

SIGN UP FOR HEROKU

[HTTPS://SIGNUP.HEROKU.COM/](https://signup.heroku.com/)

**[HTTPS://DEVCENTER.HEROKU.COM/
ARTICLES/GETTING-STARTED-WITH-
RUBY#SET-UP](https://devcenter.heroku.com/articles/getting-started-with-ruby#set-up)**

Get the heroku toolbelt

ONCE THE HEROKU TOOLBELT IS INSTALLED

- ▶ EVERYONE have it installed for the right OS?
- ▶ Go to your terminal
- ▶ heroku login

```
vincent@apple:~ $ heroku login  
Enter your Heroku credentials.  
Email: trivett@gmail.com  
Password (typing will be hidden):
```

POSTGRESQL

- ▶ Rails uses SQLite3 by default
- ▶ This is fine and all but it isn't a really industrial-strength database.
- ▶ Heroku requires you to use Postgresql, a more modern relational database.



**THIS TAKES A WHILE,
DON'T JUMP AHEAD!**

GET POSTGRESQL (MAC)

- ▶ `brew install postgresql`
- ▶ `ln -sfv /usr/local/opt/postgresql/*.plist ~/Library/LaunchAgents`
- ▶ `launchctl load ~/Library/LaunchAgents/homebrew.mxcl.postgresql.plist`

GET POSTGRESQL (UBUNTU)

- ▶ `sudo sh -c "echo 'deb http://apt.postgresql.org/pub/repos/apt/ precise-pgdg main' > /etc/apt/sources.list.d/pgdg.list"`
- ▶ `wget --quiet -O - http://apt.postgresql.org/pub/repos/apt/ACCC4CF8.asc | sudo apt-key add -`
- ▶ `sudo apt-get update`
- ▶ `sudo apt-get install postgresql-common`
- ▶ `sudo apt-get install postgresql-9.3 libpq-dev`
- ▶ `sudo -u postgres createuser <your user name> -s`

**NOBODY DOES SQUAT
UNTIL WE ARE ALL DONE
INSTALLING POSTGRESQL**

USING POSTGRESQL IN BLOGR

- ▶ We will add the right gems to the rails app so it knows which database to use and when.
- ▶ Move the gem sqlite3 into the production/test group.
- ▶ Add rails_12factor and pg to production

```
group :production do  
  gem 'pg'  
  gem 'rails_12factor'  
end
```

- ▶ `bundle install`

DATABASES + ACTIVE RECORD

- ▶ As far as you are concerned, your code that touches the model *does not need to change* to accommodate PostgreSQL
- ▶ This is awesome. Rails knows how to talk to separate different databases.
- ▶ So in development (everything we have done so far) it will write to an SQLite3 database. On Heroku, it will use PostgreSQL

CONFIG/DATABASE.YML

- ▶ This is a config file that tells rails how to talk to the correct database. Edit production like so:

```
production:
  adapter: postgresql
  encoding: unicode
  pool: 5
  database: blogr_production
  username: blogr
  password: <%= ENV[ 'BLOGR_DATABASE_PASSWORD' ] %>
```

- ▶ Here's what this means...

DEVELOPMENT/PRODUCTION PARITY

- ▶ Going forward, now that you have Postgres on your computer, you have the option of using PostgreSQL in development as well.
- ▶ You can do this with:
- ▶ `rails new appname --database=postgresql`
- ▶ If you do this, you won't have to edit `database.yml` or add the `pg` and rails 12 factor gems manually.

READY TO LAUNCH

- ▶ In Blogr:
- ▶ `heroku create`
- ▶ `git push heroku master`
- ▶ This will take a while.

WHEN IT FINISHES

- ▶ execute heroku open
- ▶ and.....

AN ERROR!

- ▶ Whooo hoo! Congratulations, your first error on the open web.
- ▶ How did we diagnose errors like this in development??

LOGS

- ▶ Run `heroku logs --tail`
- ▶ Refresh the page and watch the logs. See if you can figure out what we missed.

YOU GUESSED IT

- ▶ You need to migrate the database
- ▶ How to do this on heroku:
- ▶ `heroku run rake db:migrate`

REFRESH

- ▶ You now have a friggin app actually working on the internet.

RENAME

- ▶ Heroku generates a zen-like name for your app.
- ▶ You can rename it to whatever you like.
- ▶ Please rename your app to something like casonye-blogr
- ▶ `heroku rename casonye-blogr`
- ▶ casonye-blogr.herokuapp.com

LAB

- ▶ Get the latest iteration of your final project up on Github and Heroku.
- ▶ Push to Github, then push to heroku
- ▶ Rename it appropriately
- ▶ Send Eddie and I the URL and the link to your git repo
- ▶ Keep working on your app after you finish. We are here to help.