Chef + AWS + CodeIgniter

@_kennyk_

Me

- Co-authored Sparks
- Reactor Engineer
- Former Yahoo, Freelancing w/ Mozilla
- Closet Ops guy

Deployment Before

- Collection of bash scripts
- Not all installed software is known
- Homebrew tools (Yahoo's Taxi, Pogo, Etsy's Deployinator)
- Didn't integrate well with infrastructure
  - Provision new boxes? Probably not
- Terrifying to edit once the guy who wrote it is gone

Now

- Capistrano
- Rubber
- Puppet
- Chef
- .. and a bunch more

Why Chef?

- Integration with AWS and Rackspace
- Extremely flexible
- Platform-agnostic
  - And linux-variant agnostic
- Code sharing in the community
- Very popular
- Back by OpsCode, founded by former Amazon Ops guys

Any Downsides?

- So flexible, so open-ended, big learning curve
- 30-40 hours before you feel comfortable
  - According to me
- You'll rely on mostly blog posts to get your specific stack running
  - Or presentations like this

But just imagine …

- Your server stops taking requests

- You don't know why

- You can
  - Run one command that provisions a new box in AWS with your full stack and configuration
  - Run commands on all of your nodes at once

Uh yea, so what about Heroku/ Pagoda/PHP Fog?

- Just my opinion
- When they go down, so do you
- You trust Heroku to manage infrastructure intelligently with a black box
- But sometimes they don't, apparently
  - Heroku, June xx, whoops
- Don't be someone else's bitch

Pre-reqs

- You know a little something about AWS
- You're comfortable with the command line

Demo

By the Way

- You can't learn what you *really* need to know in a 45 minute talk

- In ruby
  - Can still use it without knowing much

Chef Terms (the basics)

- Cookbooks
- Recipes
- Knife
- Roles
- Attributes
- Environments
- Data bags

Cookbooks

- A module for installing and configuring a software package
  - PHP
  - Apache2
  - New Relic monitoring
  - Your webapp
  - …
- *A cookbook contains at least one recipe*

Recipes

- A component of each cookbook
  - For Apache2
    - mod_rewrite
    - mod_php5
    - …
- Each cookbook has a *default* recipe

Resources

- Build in tools/building blocks for writing recipes
  - Need a symlink? Got that
  - Need an apt package? Got that
- http://wiki.opscode.com/display/chef/Resources

Attributes

- Cascading configuration settings that are used by recipes
  - What port does Apache listen on?
  - What user does this process run as?
- You can make anything a setting
- Set them at the
  - Cookbook level (defaults)
  - Environment level (dev, staging, prod, w/e)
  - Any role level

Roles (rolls)

- Categories for your servers/nodes
- In sophisticated setups
  - Database server role
  - Webapp role
  - Load balancer role
  - Utility box role
- A single server can play multiple roles
  - In dev, maybe one box plays all roles to save $

Environments

- Just a container/construct for different env settings
- Each node is assigned an environment on creation
- Can be changed

Data bags

- More settings, usually used for sensitive settings
  - Database passwords
  - API keys
  - ...
  - Settings which aren't strictly associated with a role or environment type
- Can be encrypted

Actual node

- Has environment
- Has run_list
  - Usually set once on node creation
  - Recipes
  - Roles
- Has chef-client installed
  - Pulls settings
  - Figures out what's needed
  - Installs

Knife

- Command line tool to rule your world
- See what your nodes are up to
- Search, run remote commands
- Deploy at scale
  - 14,000 adserving nodes?
- Executes in parallel

Hosted Chef

- OpsCode is the central repository for your configuration

- Diagram next slide
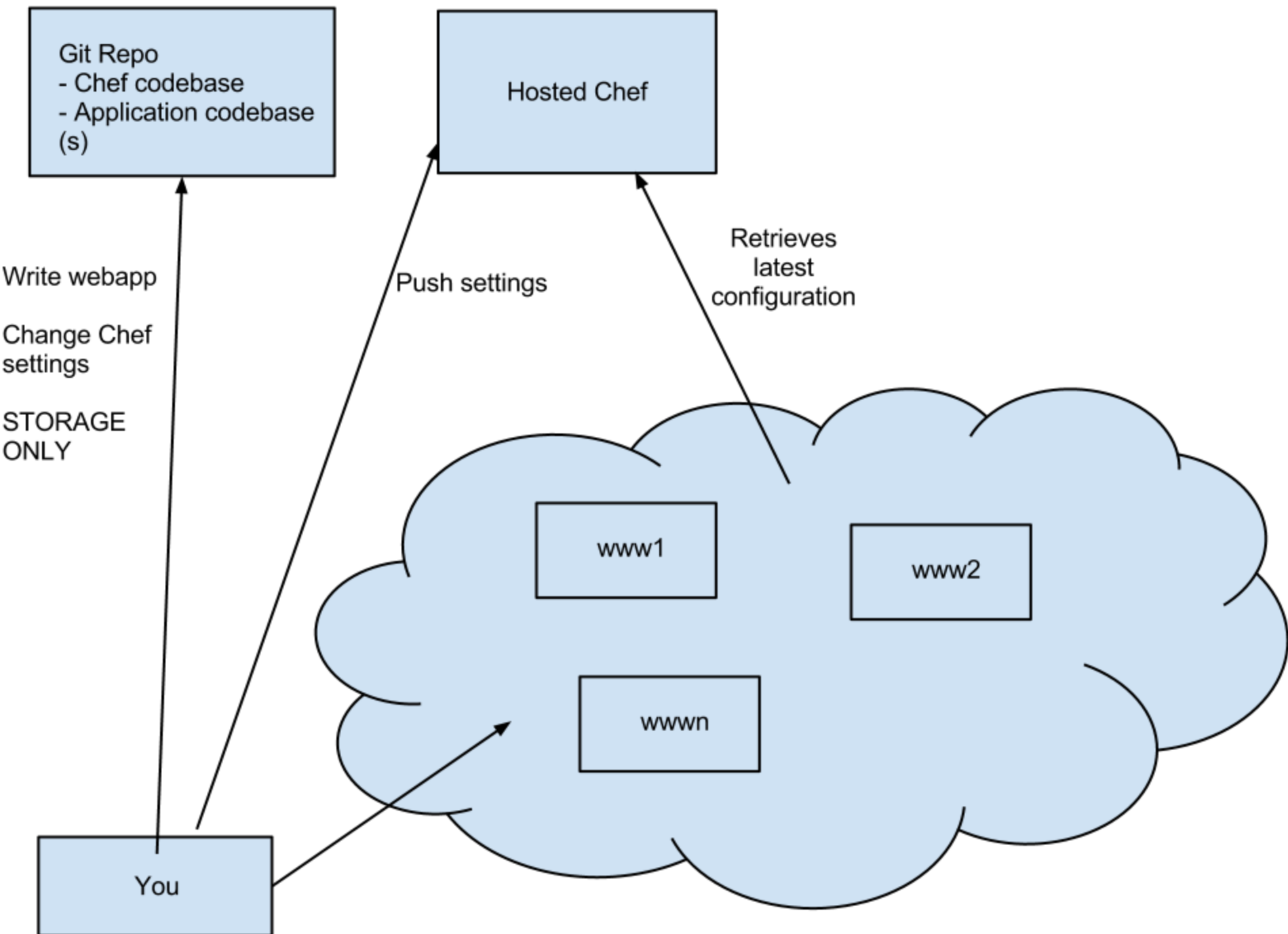
Pushing settings

- knife cookbook upload [cookbook]
- knife role from file roles/webapp.json
- knife data bag from file webapp [path]
- knife environment from file [path]

More demo

- Download a community cookbook
- Execute in paralell

Thanks!

- @_kennyk_
- codefury.net