Vagrant and Ansible

A brief and confusing intro

Filip Joelsson

filip@pixiecoder.com
Consulting Web Developer
Jack of all trades
Serial learner

Vagrant? What’s that?

Vagrant is infrastructure that you can use to get a box up and running in a predictable manner.

What's in a box?
The frozen state of a VM

When is it useful?

When you quickly need to get a predictable environment up and running

When is it extra useful?

When you quickly need to get many developers up and running in a clone of the production environment

What are the problems?

Well, it's just infrastructure. If the destination sucks, it doesn't help much to have a nice road

Where can you go?

- Common virtualization environments:
- VirtualBox (free, slow, the most common)
- VMWare (evil, but more performant)
- AWS (so yes, input you keys in the vagrant file and you can use the same config to push your stuff into production as well)

Vehicles?

”Provisioning”:

- Quick and dirty: Shell
- Active client : Chef and Puppet
- Remoting: Ansible

Provisioning examples:

1. Shell
   Pros: Quick, comprehensible
   Cons: Hard to cover everything

2. Ansible – beginner
   Write yaml for everything

3. Ansible – intermediate
   Refactor, use vars and maybe a jinja2 template

4. Ansible – advanced
   Roles, catch (almost?) everything to get drupal up and running

5. Galaxy
   Re-use other peoples stuff

Get your hands dirty

...so you can keep your dev environment clean!

Links:

https://austin2014.drupal.org/session/devops-humans-ansible-drupal-deployment-victory.html

https://adamcod.es/2014/09/23/vagrant-ansible-quickstart-tutorial.html

https://github.com/geerlingguy/ansible-vagrant-examples