

Drupal in Virtual Environments

(or whatever that means!)

mikeegoulding 

 @mikeethedude

Two things about this session...

Two things about this session...

- This isn't a how-to.

Two things about this session...

- This isn't a how-to.
- There really isn't a second thing.

Two things about this session...

- This isn't a how-to.
- There really isn't a second thing.



Two things about this session...

- This isn't a how-to.
- There really isn't a second thing.

Anyway...



Why virtualize your local
dev?

Why virtualize your local dev?

- Consistency

Why virtualize your local dev?

- Consistency
 - Multiple locations, computers, and OS on an average dev team.

Why virtualize your local dev?

- Consistency
 - Multiple locations, computers, and OS on an average dev team.
- Speed Improvements

Why virtualize your local dev?

- Consistency
 - Multiple locations, computers, and OS on an average dev team.
- Speed Improvements
 - Spin up new sites quicker

Why virtualize your local dev?

- Consistency
 - Multiple locations, computers, and OS on an average dev team.
- Speed Improvements
 - Spin up new sites quicker
 - Spin up old sites quicker.

The image shows a page from a historical Japanese document, likely a ledger or account book, written in vertical columns of cursive Japanese calligraphy (kuzushiji). The text is organized into multiple columns, with some sections featuring bolded headers or titles. The handwriting is dense and characteristic of Edo-period documents. The page number '二' (Two) is visible at the top right.

At the top left, there are several small rectangular stamps or seals. The main body of the text consists of numerous vertical entries, many of which begin with numerical values followed by descriptive text. Some entries include dates or specific names. The overall layout suggests a systematic recording of transactions or events over time.

Key features of the document include:

- Vertical orientation of all text.
- Dense cursive calligraphy (kuzushiji).
- Numerical entries interspersed with descriptive text.
- A header section at the top right containing the character '二'.

What are the
options?

What are the options?

Where do I even get started??

The environment builder things.
(The apps that make this possible.)

The environment builder things.
(The apps that make this possible.)

- Vagrant (<https://www.vagrantup.com>)
 - Works with Virtualbox and VMware.

The environment builder things. (The apps that make this possible.)

- Vagrant (<https://www.vagrantup.com>)



- Works with Virtualbox and VMware.

- Docker (<https://www.docker.com>)



- Containers and a cool whale logo!

The environment builder things. (The apps that make this possible.)

- Vagrant (<https://www.vagrantup.com>)
 - Works with Virtualbox and VMware.

- Docker (<https://www.docker.com>)



- Containers and a cool whale logo!

The environment builder things. (The apps that make this possible.)

- Vagrant (<https://www.vagrantup.com>)
 - Works with Virtualbox and VMware.
- Docker (<https://www.docker.com>)
 - Containers and a cool whale logo!

The environment builder things.
(The apps that make this possible.)

The environment builder things.
(The apps that make this possible.)

All-in-one Solutions

The environment builder things. (The apps that make this possible.)

All-in-one Solutions

- Kalabox (<http://www.kalamuna.com>)
 - Vagrant or Docker depending on the version.

The environment builder things. (The apps that make this possible.)

All-in-one Solutions

- Kalabox (<http://www.kalamuna.com>)
 - Vagrant or Docker depending on the version.
- PuPHPet (<https://puphpet.com>)
 - Uses Vagrant and the puppet provisioner.

Provisioners

(The scripts that build your environments)

Provisioners

(The scripts that build your environments)

- Puppet (<https://puppetlabs.com>)

Provisioners

(The scripts that build your environments)

- Puppet (<https://puppetlabs.com>)
- Chef (<https://www.chef.io>)

Provisioners

(The scripts that build your environments)

- Puppet (<https://puppetlabs.com>)
- Chef (<https://www.chef.io>)
- Ansible (<http://www.ansible.com>)

Provisioners

(The scripts that build your environments)

- Puppet (<https://puppetlabs.com>)
- Chef (<https://www.chef.io>)
- Ansible (<http://www.ansible.com>)
- Fabric (<http://docs.fabfile.org/en/latest>)

Provisioners

(The scripts that build your environments)

- Puppet (<https://puppetlabs.com>)
- Chef (<https://www.chef.io>)
- Ansible (<http://www.ansible.com>)
- Fabric (<http://docs.fabfile.org/en/latest>)
- Others I haven't heard of.

That seems like a lot of options...

How will I know which combination is right for
my team?

Consider

Consider

- Dedicated DevOps/IT team?

Consider

- Dedicated DevOps/IT team?
- One man/woman show or a big team?

Consider

- Dedicated DevOps/IT team?
- One man/woman show or a big team?
- How much customization do you want/need?

Pitfalls



Pitfalls



- There will be things to install locally on each machine.
- Won't magically work on each computer without a little effort.
- Troubleshooting provisioners isn't simple without a little knowledge of the parts.
- Don't Hack It!
- Syncing can be a pain.

Best Practices

(Or how to get the most out of it)

- Start with an existing project
 - Like Vampd (<https://github.com/vampd/vampd>)
 - or VLAD (<https://github.com/hashbangcode/vlad>)
- Go Slow
- If you don't like ____ language don't use a provisioner based on it.
- Mirror what matters from Prod
- Submodules!
- Research



Thanks!

Questions?

mike@ashday.com

Credit for things I found:

- pitfall image (http://upload.wikimedia.org/wikipedia/en/5/54/A2600_Pitfall.png)
- vagrant logo (<http://vagrantup.com>)
- docker logo (<http://docker.com>)
- matrix image (<http://www.consciousconsumernetwork.tv/>)
- awesome face (<http://knowyourmeme.com/memes/awesome-face-epic-smiley>)