We use a lot of VMs and containers for testing and building Samba. The setup and maintenance of these machines requires a lot of work.

How can we reduce this work while making the result more reproducible and disposable?
Enter Vagrant...

- Create and configure virtual test/dev environments:
  1. reproducible
  2. disposable
  3. lightweight
  4. portable
- very easy management
- https://www.vagrantup.com/
- Mitchell Hashimoto, hashicorp
Building Blocks

1. images of base installs (*base boxes*)
   (several backends supported for virtualization)

2. Vagrantfile to configure and further setup
   (several *provisioners* supported)

3. command `vagrant` to manage all aspects

   - ⇒ simple but very powerful setup
   - ⇒ similar to docker concepts
Providers - Included

- virtualbox
- docker ("Er, docker?..." - "Yes!" 😊)
- Hyper-V
Providers - via Plugins

**local**
- lxc
- **libvirt** (qemu/kvm)
- native kvm
- parallels
- ...

**remote**
- aws
- azure
- cloudstack
- openstack
- rackspace
- ...

---

Michael Adam  
vagrant (6/20)
Hey, we can use LXC! 😊

Yesterday:

"We can use LXC containers for this.  
- I don't know much about LXC.  
- Don't worry, It's just Docker without the hype."

Englisch übersetzen

11:29 PM · 08 Mai 15

Michael Adam  vagrant (7/20)
Providers - via Plugins - not free!

- vmware
- vmware fusion
- vmware workstation
Other Useful Plugins

- `vagrant-cachier` - cache packages on the host
- `vagrant-mutate` - convert boxes between providers
- ...

Michael Adam
vagrant (9/20)
Base Boxes

- Many on hashicorp’s atlas (atlas.hashicorp.com).
- Mostly virtualbox and vmware images.
- `libvirt`: convert vbox ⇒ libvirt with `vagrant-mutate`
- `lxc`: there are a few boxes ⇒ need to create on your own
  - I created lxc boxes for fedora
  - published: https://atlas.hashicorp.com/obnox/
  - git://git.samba.org/obnox/vagrant/
    vagrant-lxc-base-boxes.git
  - more boxes needed!
Vagrantfile

- it is a genuine ruby program

minimal example

```ruby
Vagrant.configure("2") do |config|
  config.vm.box = "hashicorp/precise32"
end
```
Control

vagrant [options] [command] [args]

- **up**: create, start and provision
- **provision**: provision the machine
- **reload**: restarts, load new config
- **suspend**: suspends the machine
- **resume**: resume a suspended machine
- **halt**: stops the machine
- **destroy**: stops and deletes the machine
- **status**: outputs status of the machine
- **ssh**: connects to machine via SSH
- **plugin**: manages plugins
- **box**: manages boxes
Config Management

- called provisioners
- supported:
  1. shell (inline/external)
  2. ansible
  3. puppet
  4. chef
  5. ...
Notes about Installation

- **Upstream**: current packages at vagrantup.com
  - version 1.7.2
  - installed, under /opt
  - ships ruby and many gems
  - sometimes problems with system-ruby
- **Ubuntu** has older vagrant package
  - version 1.6.5
  - no additional plugins shipped
- **Fedora (≥ 21)** has recent vagrant package now
  - version 1.7.2
  - package for vagrant-lxc
  - package for vagrant-libvirt
Demo Time (vagrant)
Enter VaSaBi ...
- Vagrant Samba Build ... 😊
- A tool to build and selftest Samba on various platforms with a single command
- Making use of vagrant
- Including an out-of-tree samba build wrapper (using symlinks)

  git://git.samba.org/obnoxx/vasabi.git
Demo Time (VaSaBi)
TODOs

- let vasabi create and manage the Vagrantfile (under `~/.vasabi`)
- dynamically let the vasabi script manage machines based on
  1. base box
  2. source directory (hash thereof)
  3. possibly additional build ID
- save test results (`st/`) like `bin/`
- add support for more boxes
- possibly add support for autobuild-like jobs instead of plain `make test`
Thanks for your attention!

Questions?

obnox@samba.org
madam@redhat.com