

Using the -no-provision directive for Vagrant when resuming machines

It seems that there is an active issue when resuming machines using Vagrant that triggers provisioning scripts on resume and not just when doing the original `vagrant up` command.

EXAMPLE:

```
Eric's-MacBook-Pro-2:virtualbox-docker-sandbox DiscoPosse$ vagrant resume dockersandbox
=> dockersandbox: Resuming suspended VM...
=> dockersandbox: Booting VM...
=> dockersandbox: Waiting for machine to boot. This may take a few minutes...
dockersandbox: SSH address: 127.0.0.1:2222
dockersandbox: SSH username: vagrant
dockersandbox: SSH auth method: private key
dockersandbox: Warning: Remote connection disconnect. Retrying...
dockersandbox: Warning: Remote connection disconnect. Retrying...
=> dockersandbox: Machine booted and ready!
=> dockersandbox: Running provisioner: shell...
dockersandbox: Running: /var/folders/2m/hrdft32956j6n0r0d_dr9g4r0000gn/T/vagrant-shell20160601-16217-two1lv.sh
=> dockersandbox: stdin: is not a tty
=> dockersandbox: 0 value set
=> dockersandbox: Ign http://us.archive.ubuntu.com trusty InRelease
=> dockersandbox: Get:1 http://us.archive.ubuntu.com trusty-updates InRelease [65.9 kB]
=> dockersandbox: Ign http://apt.puppetlabs.com trusty InRelease
=> dockersandbox: Hit http://apt.puppetlabs.com trusty Release.gpg
```

The issue here is that I already have the build script executed on the original vagrant build of the machine. The scripts may not be idempotent, and could overwrite content or damage the active machine.

In our Vagrant file, we use the provision capability regularly, so we would not want to have to build all sorts of logic around that unless necessary because Vagrant did this natively in the past.

```
docker sandbox.vm.box = "trusty-server"
docker sandbox.vm.box_url = "https://oss-binaries.phusionpassenger.com/va
docker sandbox.vm.network :private_network, ip: "10.180.0.30", :netmask =
docker sandbox.vm.provision :shell, :path => "build.sh"
end
end
```

Workaround Using the -no-provision Parameter

Rather than run a `vagrant resume` as you saw above which triggered the build script again, you can simply use a `vagrant up --no-provision` which will bring the machine up and reconnect any SSH connections and NFS shares, but it will ignore any provision directives from the Vagrantfile:

```
connection to 127.0.0.1 closed.
Eric's-MacBook-Pro-2:virtualbox-docker-sandbox DiscoPosse$ vagrant suspend dockersandbox
=> dockersandbox: Saving VM state and suspending execution...
Eric's-MacBook-Pro-2:virtualbox-docker-sandbox DiscoPosse$ vagrant up --no-provision
Bringing machine 'dockersandbox' up with 'virtualbox' provider...
=> dockersandbox: Resuming suspended VM...
=> dockersandbox: Booting VM...
=> dockersandbox: Waiting for machine to boot. This may take a few minutes...
dockersandbox: SSH address: 127.0.0.1:2222
dockersandbox: SSH username: vagrant
dockersandbox: SSH auth method: private key
dockersandbox: Warning: Remote connection disconnect. Retrying...
dockersandbox: Warning: Remote connection disconnect. Retrying...
dockersandbox: Warning: Remote connection disconnect. Retrying...
dockersandbox: Warning: Remote connection disconnect. Retrying...
dockersandbox: Warning: Remote connection disconnect. Retrying...
dockersandbox: Warning: Remote connection disconnect. Retrying...
dockersandbox: Warning: Remote connection disconnect. Retrying...
dockersandbox: Warning: Remote connection disconnect. Retrying...
dockersandbox: Warning: Remote connection disconnect. Retrying...
=> dockersandbox: Machine booted and ready!
=> dockersandbox: Machine not provisioned because `--no-provision` is specified.
Eric's-MacBook-Pro-2:virtualbox-docker-sandbox DiscoPosse$ █
```

Hopefully this will be solved in a patch or future update to Vagrant. The post deals specifically with version 1.8.1 that presented the problem. It may also be present in other versions.