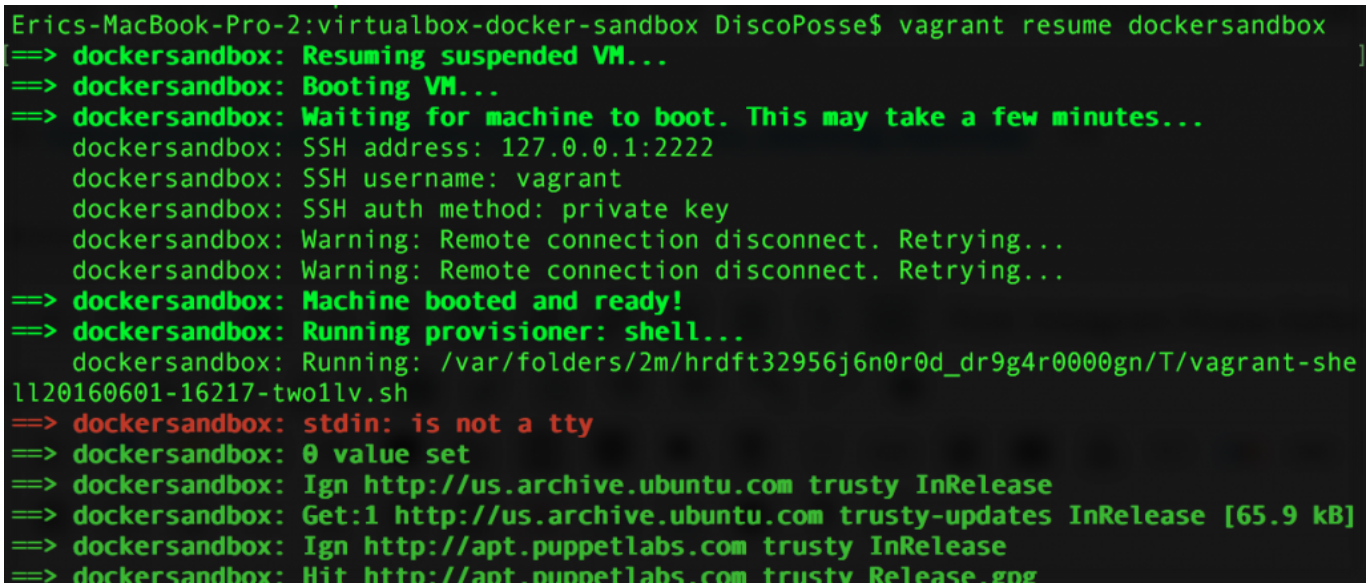
Using the –no-provision directive for Vagrant when resuming machines

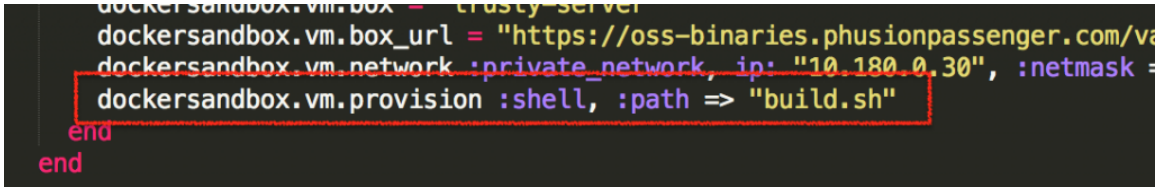
It seems that there is an active issue when resuming machines using Vagrant that triggers provisioning scripts on resume and not just when doing the original `vagrant up` command.

EXAMPLE:



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.



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:

```
Erics-MacBook-Pro-2:virtualbox-docker-sandbox DiscoPosse$ vagrant suspend dockersandbox
==> dockersandbox: Saving VM state and suspending execution...
Erics-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.
Erics-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.

PowerCLI – Enabling or Disabling VAAI in a vSphere Cluster

There are limited cases where you would need to do this, but since I'm a huge proponent for scripting anything I do, here is a programmatic (and quick) way to enable or disable VAAI functionality for your vSphere clusters.

Why Disable VAAI?

The first thing you are asking of course, is why on earth would we do this? In vSphere 5.0, there is a bug affecting VAAI enabled systems that could cause a PSOD (Purple Screen of Death). The workaround for this is to upgrade to 5.1 where the problem no longer exists. There may be a patch for 5.0 which comes, but at this point it is still an outstanding issue.

For those of us who can't jump to the next version at the moment, there is a simple way to disable the VAAI functions which will trigger the PSOD.

Disabling VAAI in the vSphere Client

This can be done in the vSphere Client also, but when you have larger clusters or multiple clusters it can get a little tedious (and error prone) to do this in the GUI. We access the VAAI attributes by opening up the **Configuration** tab for our host and selecting **Advanced Settings**:

Once you are in the Advanced Settings configuration dialog, select the **DataMover** option in the left hand pane and you will see the two fields in the main window that are **HardwareAcceleratedMove** and **DataMover.HardwareAcceleratedInit**:

The option are 0 for disabled or 1 for enabled. One more setting to configure under the **VMFS3** option which is **VMFS3.HardwareAcceleratedLocking**:

NOTE: the settings take effect immediately and do not require a reboot of the host. As soon as you close your Advanced Settings window you will see the task in your vSphere client window.

As I've mentioned, this isn't terribly difficult, but we may have numerous hosts, and maybe even numerous clusters to manage so that is where the magic of PowerCLI comes in.

Script all the things!

My script is simple, and does the following:

1. Check the current value of the settings for all hosts in the cluster (replace **YourClusterName** in the script)
2. Change the properties to be 0 for disabled
3. Check the current value of the three settings to confirm the new results

So here it is:

Not much, but the good thing is that once you have the values set to 0, you can simply adjust the script to set them to 1 again without much effort at all.

Here is the script in text format for easier copy and paste. Hope it's helpful!

```
# First let's see the current setting to ensure it is set to 1 (enabled) Get-Cluster
YourClusterName | Get-VMHost | Get-AdvancedSetting -Name
DataMover.HardwareAcceleratedMove Get-Cluster YourClusterName | Get-VMHost |
Get-AdvancedSetting -Name DataMover.HardwareAcceleratedInit Get-Cluster
YourClusterName | Get-VMHost | Get-AdvancedSetting -Name
VMFS3.HardwareAcceleratedLocking

# Now we disable the three options by setting the value to 0 Get-Cluster
YourClusterName | Get-VMHost | Set-VMHostAdvancedConfiguration -Name
DataMover.HardwareAcceleratedMove -Value 0 Get-Cluster YourClusterName | Get-
VMHost | Set-VMHostAdvancedConfiguration -Name
DataMover.HardwareAcceleratedInit -Value 0 Get-Cluster YourClusterName | Get-
VMHost | Set-VMHostAdvancedConfiguration -Name
VMFS3.HardwareAcceleratedLocking -Value 0

# Confirm the results by querying the values which will show 0 now (disabled) Get-
Cluster YourClusterName | Get-VMHost | Get-AdvancedSetting -Name
```

```
DataMover.HardwareAcceleratedMove Get-Cluster YourClusterName | Get-VMHost |
Get-AdvancedSetting -Name DataMover.HardwareAcceleratedInit Get-Cluster
YourClusterName | Get-VMHost | Get-AdvancedSetting -Name
VMFS3.HardwareAcceleratedLocking
```

Happy Scripting!