Table of Contents

DIY Private Cloud ... using VirtualBox and Chef

Introduction + About Me

Hello, my name is Wil Moore III

I work for a US-based company by the name of Net-Results. Our main product is a marketing automation platform.

We are located in Denver, Colorado.

A TOOLKIT for infrastructural AUTOMATION

* THE BACKSTORY *

About a year ago, I worked for a company in the HealthCare industry.

We were dealing with HIPAA, SOX, etc., etc.

Amazon Web Services looked attractive but we couldn't get approval.

Because we were unable to utilize the public cloud, we had to attempt to achive our scalability and elasticity within our own firewall.

We also wanted the freedom to experiment with new technologies and be able to throw things away at will.

Another major goal was to be able to understand a year later why we made certain infrastructural decision. Everything needed to be self-documenting and we needed an audit trail.

High Quality Free Components Everywhere

My experience has been that open-source tools are *generally* higher quality than their non-open equivalents.

Open generally means less BS and more bug fixes.

Security holes get reported and corrected quicker.

No silver bullet

- there is no one size fits-all model
- your infrastructure and available resources are unique

WHY VIRTUALBOX?

- We started w/ Virtualbox because we had just switched over from VMWare for our local development.
- I started prototyping and it turns out that Virtualbox is quite up the the task of allowing a smooth automation path.
- Don't cargo-cult when you hear something isn't fast or doesn't scale. It may suit you just fine.

- It's Available for Windows, Mac, and Linux.
- It supports a bunch of image formats out of the box.

That being said, don't let me force you into using Virtualbox. XEN and KVM are rediculously good and could actually be a better choice once you understand how it all works.

KEYS to AUTOMATION

- Going forward we will look at several ways to automate critical pieces of the puzzle.
- The goal is to hand you the keys so you can go back to your organization and drive your own automation effort.

LATEST VERSION PLEASE?

- You want to automate as much as possible...even the host machine installations.
- Script your VM software install so it becomes painless. Do this on your host machines as well as your development machines.
- In order to be lazy, you have to prepare for it.

Choose an API/interface

- There are several options, but let me save you some time. The CLI interface works well and is available cross-platform.
- But if you like XML, COM, or XPCOM, I won't stop you.

VboxManage sub-commands

- Remotely control the instance.
- Create VM images, start, stop, pause instances.
- Clone Images, set metadata on an instance, or pull metrics.

Cooking with Chef

- Allows you to compose your infrastructure as code
- Chef is all about /he cookbook
- Cookbooks have recipes
- Recipes have files, templates, and more
- Run-Lists determine recipes to install

Running Chef until we

- sudo chef-solo -c provisioners/chef/bin/solo.rb # Based on error, we need to create a cookbook
- sudo chef-solo -c provisioners/chef/bin/solo.rb
- No more errors; however, notice "Run List expands to []" # We need to add a run list
- A run list is a manifest of which recipes to install

Want a green environment

- In order to have a stand-by ready, but powered-down, you'll want to use a Network Power Switch.
- They are meant to be contolled remotely; however, this can also be automated.

Use ONLY what you need!

- deploy only the hardware that is needed
- preallocate a fraction of what you are using as stand-by

NOTES:

- there is no one-size-fits-all solution to figuring out how many stand-bys to use; however, mine is simple so you can start there and modify as necessary.
- in all honesty, I generally feel safer with AT LEAST 2 stand-bys as a bare minimum.