Packer

By James Reynolds

Packer, Vagrant, Docker, Kubernetes, etc etc etc

- Packer - Creates VM images (can be used in production)
- Vagrant - Uses VM images for local rapid reliable testing (for developers, not production)
- Docker - Wraps Linux and Windows software and dependencies into containers that run in isolation (similar to VM’s but without a kernel)
- Kubernetes (k8s) - Runs containers on multiple servers to automate deployment, scaling, and management
- Anka - Virtualized macOS for iOS developers, not production

Why Developer and not Production?

- Radmind as example
  - The project still exists and I find have done a few updates
  - I merged the last GitHub pull request
    - It solved a problem on one OS and created another on another OS, so I reverted the merge
    - If I had a VM setup with Mac and multiple Linux VM’s I can instantly build on each platform and run test
    - I would’ve seen this error before I merged the request

Packer

- By Hashicorp (same people who make Vagrant)
- Command line utility (open source, written in Go)
- Builds Virtual Machine images from installers
- Uses a text file “template” (JSON format)
- Runs on Mac, Windows, Linux
- Builds Mac, Windows, Linux images

Packer

- Packer works with a VM app (VMware, VirtualBox, etc)
- Packer creates a VM, assigns an installer image, boots, and uses VNC to type the “boot_command” to install the OS
- For cloud images it uses prebuilt vendor images
- The end result is a fully installed and configured VM

Packer

- The documentation is short and easy to read
- I’m covering the most important stuff in this presentation
- The hard part of Packer isn’t Packer
- The hard part is that you have to know the OS (more later)

JSON Template File

```
{
  "variables": {
    "access_key": "{{env `AWS_ACCESS_KEY_ID`}}",
    "secret_key": "{{env `AWS_SECRET_ACCESS_KEY`}}"
  },
  "builders": [{
    "type": "amazon-ebs",
    "access_key": "{{user `access_key`}}",
    "secret_key": "{{user `secret_key`}}",
    "region": "us-east-1",
    "source_ami": "ami-af22d9b9",
    "instance_type": "t2.micro",
    "ssh_username": "ubuntu",
    "ami_name": "packer-example {{timestamp}}"
  }]
}
```

Parts of Template

- Builders (only required part)
- Provisioners
- Post-Processors
- Variables and functions

Builders

- Creates the VM settings (RAM, CPU's, Hard disk size, etc)
- Specifies the input and output
  - Inputs: Cloud image, ISO
  - Outputs: EC2, Azure, CloudStack, DigitalOcean, Docker, Hyper-V, Parallels, QEMU, Vagrant, VirtualBox, VMware, etc
- (Builders are named after the target output and each target has it's own settings)

Provisioners

- Install extra software and configure the system
- Provisioners:
    - Shell script, Ansible, Chef, Converge, File, PowerShell, Puppet, Salt, etc

Post-Processors

- What to do after the image is built

- Compress files, upload image, checksum image, shell script, Vagrant, vSphere, etc

Variables and Functions

- Template functions look like: “{{function_name}}”
    - build_name, build_type, env, isotime, lower, pwd, sed, split, template_dir, timestamp, uuid, upper, user, packer_version, clean_resource_name
- Template variables look like: “{{.Variable}}”
    - .Vars, .Path, .Name, more
- User variables look like: “{{user `variable`}}”
    - Defined by user, can use external file

Template File - EC2

```
{
  "variables": {
    "access_key": "{{env `AWS_ACCESS_KEY_ID`}}",
    "secret_key": "{{env `AWS_SECRET_ACCESS_KEY`}}"
  },
  "builders": [{
    "type": "amazon-ebs",
    "access_key": "{{user `access_key`}}",
    "secret_key": "{{user `secret_key`}}",
    "region": "us-east-1",
    "source_ami": "ami-af22d9b9",
    "instance_type": "t2.micro",
    "ssh_username": "ubuntu",
    "ami_name": "packer-example {{timestamp}}"
  }]
}
```

Template File - VMware

```
"builders": [{
    "boot_command": [...],
    "disk_size": "65536",
    "floppy_files": [ "/Users/james/Jamf/http/preseed.cfg}" ],
    "guest_os_type": "ubuntu-64",
    "headless": false,
    "http_directory": "/Users/james/Jamf/http",
    "iso_checksum_type": "sha256",
    "iso_checksum": “a2cb36dc010d98ad9253ea5ad5a07fd6b409e3412c48f1860536970b073c98f5”,
    "iso_urls": [ “/Users/james/Jamf/ubuntu-18.04.2-server-amd64.iso", ],
    "output_directory": "output-jamf-vmware-iso",
    "shutdown_command": "echo 'p@s5w0rD'|sudo -S shutdown -P now",
    "ssh_password": “p@s5w0rD",
    "ssh_username": "james",
    "ssh_wait_timeout": "10000s",
    "tools_upload_flavor": "linux",
    "type": "vmware-iso",
    "vm_name": "jamf}"
}]
```

Where to get templates

- packer.io/community-tools.html
- github.com/chef/bento (incl. 10.10-10.12 & Windows)
- github.com/boxcutter (incl. 10.7-10.12 & Windows)
- github.com/geerlingguy

Running Packer

- packer validate jamf.json
- packer build jamf.json
- About 10-15 minutes to finish
- Easy!
- …
- ?

DETOUR

Sheena Iyengar's jam study

- 24 choices
  - 60% of shoppers
  - 3% purchased (1.8%)
- 6 choices
  - 40% of shoppers
  - 30% purchased (12%)

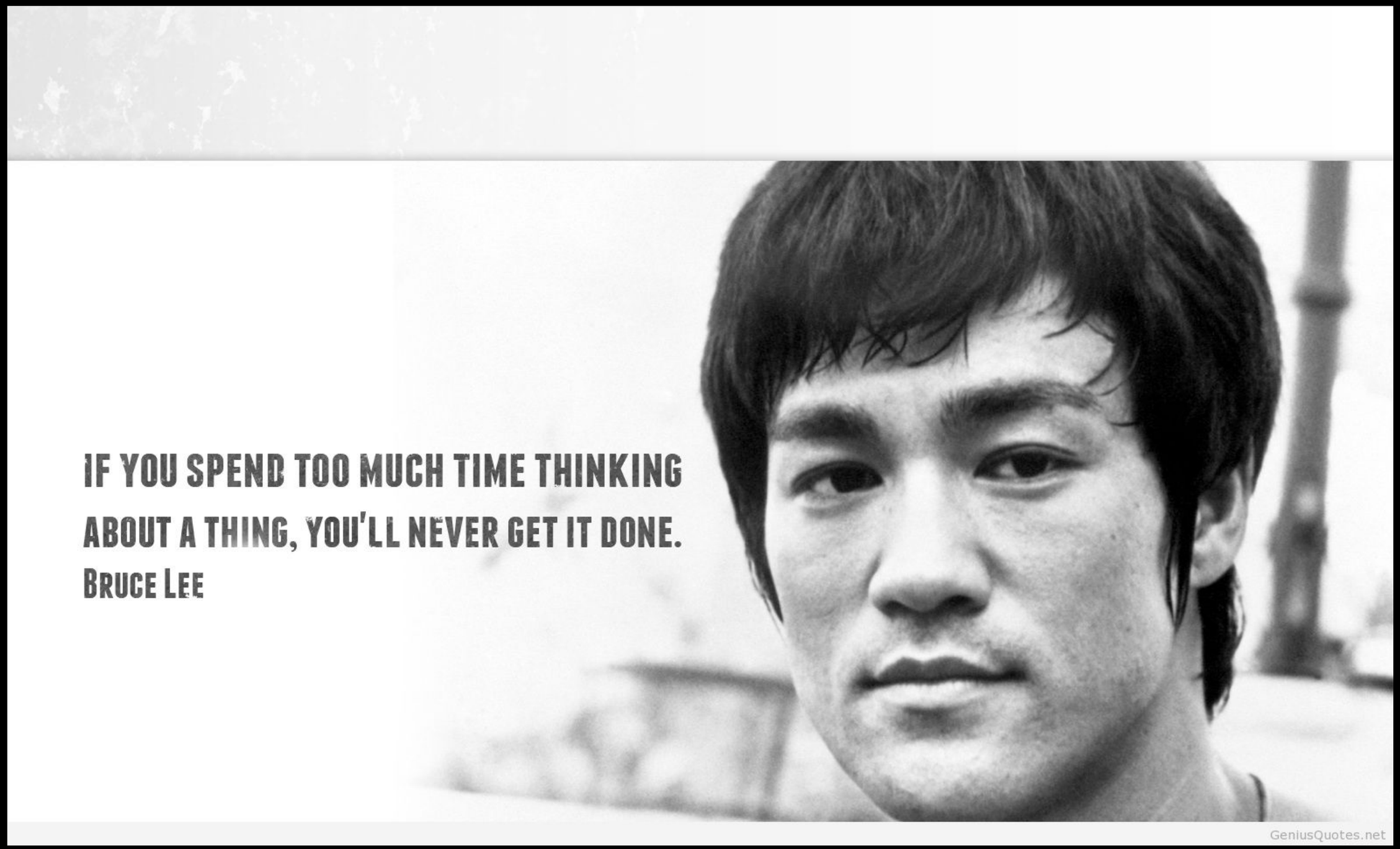
- Further studies showed that more choices also produced less satisfaction after making a choice

Infants on Life Support

- In the US the parents choose if/when to end life support
- In France the doctors choose if/when to end life support
- Parents in the US were more depressed, confused, angry, self-condemning, and had more on negative “what-if’s”
- Parents in the US wanted to choose more than letting doctors choose

Informed vs Uniformed

- The issue is between informed choosers and uniformed choosers
- And the time/resources it takes to become informed
- E.g. electronic parts from Amazon vs Adafruit/ Sparkfun/etc
    - Adafruit/Sparkfun/etc vet what they sell


IF YOU SPEND TOO MUCH TIME THINKING
ABOUT A THING, YOU'LL NEVER GET IT DONE.
BRUCE LEE
GeniusQuotes.net

Installing Mac vs Linux

- Remember I said Packer isn’t the hard part?
- The Ubuntu Installation Guide is 144 pages long…
- Luckily you (probably) only need to read about 25 pages

But...

The boot command

```
"builders": [{
    "boot_command": [...],
    "disk_size": "65536",
    "floppy_files": [ "/Users/james/Jamf/http/preseed.cfg}" ],
    "guest_os_type": "ubuntu-64",
    "headless": false,
    "http_directory": "/Users/james/Jamf/http",
    "iso_checksum_type": "sha256",
    "iso_checksum": “a2cb36dc010d98ad9253ea5ad5a07fd6b409e3412c48f1860536970b073c98f5”,
    "iso_urls": [ “/Users/james/Jamf/ubuntu-18.04.2-server-amd64.iso", ],
    "output_directory": "output-jamf-vmware-iso",
    "shutdown_command": "echo 'p@s5w0rD'|sudo -S shutdown -P now",
    "ssh_password": “p@s5w0rD",
    "ssh_username": "james",
    "ssh_wait_timeout": "10000s",
    "tools_upload_flavor": "linux",
    "type": "vmware-iso",
    "vm_name": "jamf}"
}]
```

Debian boot command

```
"boot_command": [
   "<esc><esc><enter><wait>",
   "/install/vmlinuz noapic ",
   "initrd=/install/initrd.gz ",
   "file=/floppy/preseed.cfg ",
   "debian-installer=en_US auto locale=en_US kbd-chooser/method=us ",
   “hostname=jamf ",
   "grub-installer/bootdev=/dev/sda<wait> ",
   "fb=false debconf/frontend=noninteractive ",
   "keyboard-configuration/modelcode=SKIP keyboard-configuration/layout=USA ",
   "keyboard-configuration/variant=USA console-setup/ask_detect=false ",
   "passwd/user-fullname=james ",
   "passwd/user-password=p@s5w0rD ",
   "passwd/user-password-again=p@s5w0rD ",
   "passwd/username=james ",
   "-- <enter>"
],
```

Ubiquity vs Debian Installer

```
"builders": [{
    "boot_command": [...],
    "disk_size": "65536",
    "floppy_files": [ "/Users/james/Jamf/http/preseed.cfg}" ],
    "guest_os_type": "ubuntu-64",
    "headless": false,
    "http_directory": "/Users/james/Jamf/http",
    "iso_checksum_type": "sha256",
    "iso_checksum": “a2cb36dc010d98ad9253ea5ad5a07fd6b409e3412c48f1860536970b073c98f5”,
    "iso_urls": [ “/Users/james/Jamf/ubuntu-18.04.2-server-amd64.iso", ],
    "output_directory": "output-jamf-vmware-iso",
    "shutdown_command": "echo 'p@s5w0rD'|sudo -S shutdown -P now",
    "ssh_password": “p@s5w0rD",
    "ssh_username": "james",
    "ssh_wait_timeout": "10000s",
    "tools_upload_flavor": "linux",
    "type": "vmware-iso",
    "vm_name": "jamf}"
}]
```

Preseed

```
"builders": [{
    "boot_command": [...],
    "disk_size": "65536",
    "floppy_files": [ "/Users/james/Jamf/http/preseed.cfg}" ],
    "guest_os_type": "ubuntu-64",
    "headless": false,
    "http_directory": "/Users/james/Jamf/http",
    "iso_checksum_type": "sha256",
    "iso_checksum": “a2cb36dc010d98ad9253ea5ad5a07fd6b409e3412c48f1860536970b073c98f5”,
    "iso_urls": [ “/Users/james/Jamf/ubuntu-18.04.2-server-amd64.iso", ],
    "output_directory": "output-jamf-vmware-iso",
    "shutdown_command": "echo 'p@s5w0rD'|sudo -S shutdown -P now",
    "ssh_password": “p@s5w0rD",
    "ssh_username": "james",
    "ssh_wait_timeout": "10000s",
    "tools_upload_flavor": "linux",
    "type": "vmware-iso",
    "vm_name": "jamf}"
}]
```

The preseed file

```
choose-mirror-bin mirror/http/proxy string
d-i base-installer/kernel/override-image string linux-server
d-i clock-setup/utc boolean true
d-i clock-setup/utc-auto boolean true
d-i finish-install/reboot_in_progress note
d-i grub-installer/only_debian boolean true
d-i grub-installer/with_other_os boolean true
d-i partman-auto/disk string /dev/sda
d-i partman-auto-lvm/guided_size string max
d-i partman-auto/choose_recipe select atomic
d-i partman-auto/method string lvm
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm boolean true
d-i partman-lvm/confirm_nooverwrite boolean true
d-i partman-lvm/device_remove_lvm boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true
d-i partman/confirm_write_new_label boolean true
d-i pkgsel/include string openssh-server cryptsetup build-essential libssl-dev libreadline-dev zlib1g-dev linux-source dkms nfs-common
d-i pkgsel/install-language-support boolean false
d-i pkgsel/update-policy select none
d-i pkgsel/upgrade select full-upgrade
d-i time/zone string UTC
tasksel tasksel/first multiselect standard, ubuntu-server

d-i console-setup/ask_detect boolean false
d-i keyboard-configuration/layoutcode string us
d-i keyboard-configuration/modelcode string pc105
d-i debian-installer/locale string en_US.UTF-8

# Create user account.
d-i passwd/user-fullname string james
d-i passwd/username string james
d-i passwd/user-password password p@s5w0rD
d-i passwd/user-password-again password p@s5w0rD
d-i user-setup/allow-password-weak boolean true
d-i user-setup/encrypt-home boolean false
d-i passwd/user-default-groups vagrant sudo
d-i passwd/user-uid string 900
```

The preseed file

- tasksel tasksel/first multiselect
  - ubuntu-desktop
  - ubuntu-server
- d-i pkgsel/include string
  - openssh-server
  - virtualbox
  - git
  - etc

How to learn all this stuff

- Ubuntu Installation Guide
- 4.6 Automatic Installation
- 5.3.2 Ubuntu Installer Parameters
- Appendex B Automating the installation using preseeding
- sudo apt-get install debconf-utils
- sudo debconf-get-selections --installer > preseed.cfg

CentOS

- CentOS uses Kickstart and it’s similar to the preseed file
- Ubuntu can use Kickstart also but it’s slightly different
- Why did I show Ubuntu?
  - Jamf install instructions specify Ubuntu
  - Otherwise I choose CentOS, which is suppose to be the better server OS
  - Learn both CentOS and Ubuntu (this stuff isn’t that hard)

Installing Jamf

- https://github.com/magnusviri/packer-templates/tree/master/jamf-10.12
- Download the Ubuntu ISO
- Download the Jamf Pro installer

Installed with preseed.cfg

- openssh-server
- openjdk-8-jdk
- mysql-server
- wget
- open-vm-tools

Shell script

- Configures the MySQL
    - Creates the jamfsoftware database and user
- Runs the Jamf Pro installer

What to change

- Change the username (unless your name is “james”)
- Network settings: 01-netcfg.yaml
- The hostname

Creating the image

- cd jamf-10.12
- packer validate jamf.json
- packer build jamf.json

Post Packer

- Start VM and login
  - `ip addr show`
- https://192.168.89.123:8443
- Change passwords
- mysql_secure_installation
- Tomcat Cert
- Import database

Questions?