TESTING YOUR LOCAL PKGSRC DEVELOPMENT

WING COMMANDER

Thomas Merkel

drscream@edgebsd.org

2017-07-01

AGENDA

- who?

- why?

- what?

WHO

- Server Ninja at SkyLime GmbH

- BashOps :-)

- Focus on automation, monitoring, system operation

- Around 800 servers to maintain

- SmartOS and a little bit pkgsrc, EdgeBSD, FreeBSD

WHY

- pkgsrc has multi-platform and OS support

- It's hard to build and test packages on all platforms

- Big effort to install "all" operating systems and configure your pkgsrc development environment
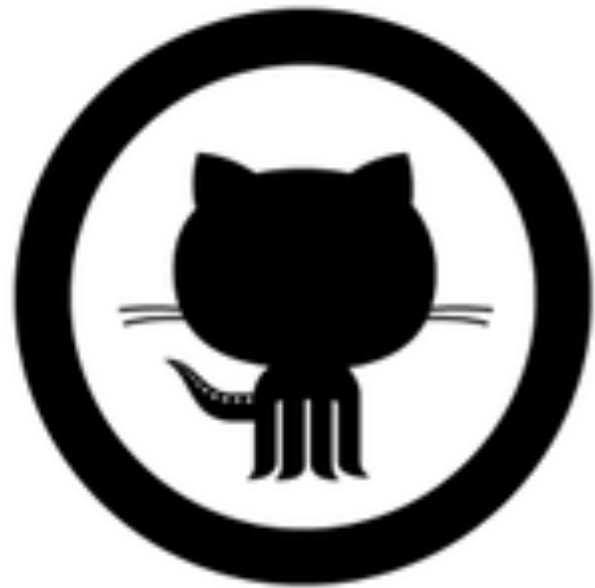
- Like to work on your local computer or notebook

WHY NOT AUTOMATE IT?

WHAT
PACKER

- "Infrastructure as code": JSON description file

- Build images for different operating systems and platforms

- Result is always an OS image

Template Code

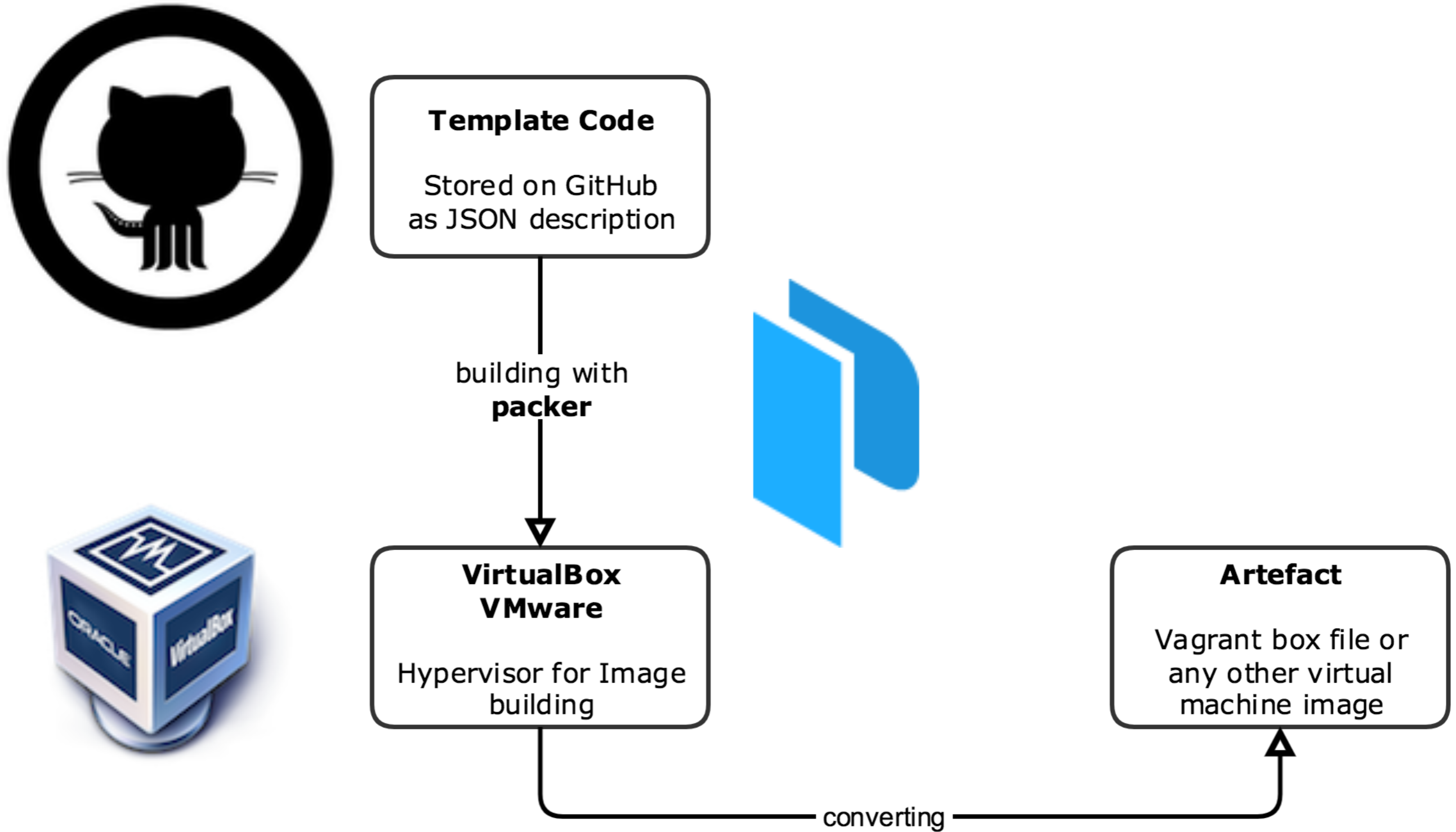Stored on GitHub
as JSON description

building with
packer

**VirtualBox
VMware**

Hypervisor for Image
building

Artefact

Vagrant box file or
any other virtual
machine image

converting

WHAT

- Support multiple builders

  - VirtualBox, VMware, QEMU

  - Google Cloud, AWS, Azure

  - and more

WHAT

- Provisioners to configure the operating system

  - Ansible, Chef, Puppet, Salt

  - Shell, PowerShell

  - and more

WHAT

- Post-Processors

  - Compress, Checksum, Manifests

  - Cloud publish, Shell, Vagrant

  - and more …

WHAT

- Define some variables

```
variables : {
  build_suffix : "",
  mirror: https://cdn.netbsd.org/pub/NetBSD,
  mirror2: http://nyftp.netbsd.org/pub/NetBSD-daily,
  release : 7.1
}
```

- Use these variables

```
iso_urls: [
  iso/NetBSD-{{user `release`}}-amd64.iso,
  {{user `mirror`}}/NetBSD-{{user `release`}}/iso/NetBSD-…-amd64.iso
]
```

WHAT

PACKER - EXAMPLE

```
"builders": [
  {
    "type": "virtualbox-iso",
    "boot_command": [
      "<enter><enter>e<enter><enter>dhcpcd wm0<enter>sleep 2<enter>",
      "ftp -o /tmp/i.sh http://{{ .HTTPIP }}:{{ .HTTPPort }}/install.sh<enter>",
      "sh /tmp/i.sh<enter>reboot<enter>"
    ],
    "disk_size": 10140,
    "guest_os_type": "NetBSD_64",
    "http_directory": "http",
    "iso_urls": [
      "iso/NetBSD-{{user `release`}}-amd64.iso",
      "{{user `mirror`}}/NetBSD-{{user `release`}}/iso/NetBSD-...-amd64.iso"
    ],
    "ssh_username": "vagrant"
  }
],
"provisioners": [
  {
    "scripts": [ "scripts/postinstall.sh" ],
    "type": "shell"
  }
]
```

DEMO

USE PACKER TO BUILD NETBSD 7 BOX

WHAT

- Local development environment

- Vagrant boxes build by packer

- Virtual machine management

- Ruby configuration file

WHAT
VAGRANT

- The configuration file describe:

  - which image should be used

  - which provider should be used

  - how to configure the network

  - if files or folders should be shared

  - additional provision scripts

WHAT
VAGRANT

- Vagrant support multiple providers

  - Default is Virtualbox

  - VMware, Hyper-V, etc. also supported

WHAT
VAGRANT

- Provisioners could be:

  - Ansible, Chef, Puppet, Salt

  - Shell scripts, Files, …

WHAT

VAGRANT - EXAMPLE

```ruby
$pkgsrc = File.expand_path("../pkgsrc", __FILE__)

Vagrant.configure("2") do |config|
  config.vm.define "netbsd7" do |netbsd7|
    netbsd7.vm.box = "builds/netbsd-7.0.1-amd64-virtualbox.box"
    netbsd7.vm.box_check_update = false
    netbsd7.vm.network :private_network, ip: "10.11.12.13"

    netbsd7.vm.synced_folder ".", "/vagrant", nfs: true
    netbsd7.vm.synced_folder $pkgsrc, "/content/pkgsrc",
                      :nfs => true,
                      :mount_options => ['nolock,vers=3,udp,noatime']

    netbsd7.vm.provider "virtualbox" do |vb|
        vb.memory = "1024"
    end

    netbsd7.vm.provision "file", source: "mk.conf.netbsd7",
                          destination: "/etc/mk.conf"

    netbsd7.vm.provision "shell", inline: <<-SHELL
        ( uptime 2> /dev/null || /bin/true )
    SHELL
  end
end
```

DEMO

START NETBSD7 MACHINE WITH VAGRANT

WHAT

- MacOS for local pkgsrc development

- Packer to build NetBSD 7 and SmartOS boxes

- Vagrant to configure and manage machines

  - Link local pkgsrc repository to these machines

- Bash scripts to run "make" commands on all boxes

WHAT

- switch.sh: Bash environment for pkgsrc development

```
# New export PATH
export PATH=/Users/tm/pkg/sbin:/Users/tm/pkg/bin:/usr/bin:/bin:/usr/sbin:/sbin

# Modify PS1
export PS1="{\!} \[\e[1;37m\]\w\[\033[m\]\[\e[32;1m\]$\[\033[m\] "

# Switch mode
bash --rcfile /Users/tm/Development/pkgsrc/.bashrc
```

- .bashrc: Use vagrant command to SSH to all boxes

```
function vmake() {
    p=$(pwd | sed 's:.*/pkgsrc/::g')
    /usr/local/bin/vagrant ssh -c "cd /content/pkgsrc/$p && sudo make $@"
}
```

DEMO

BUILD SOMETHING :-)

QUESTIONS?

https://github.com/drscream/pkgsrc-dev-env