



Modern tooling to assist with  
developing applications on  
FreeBSD



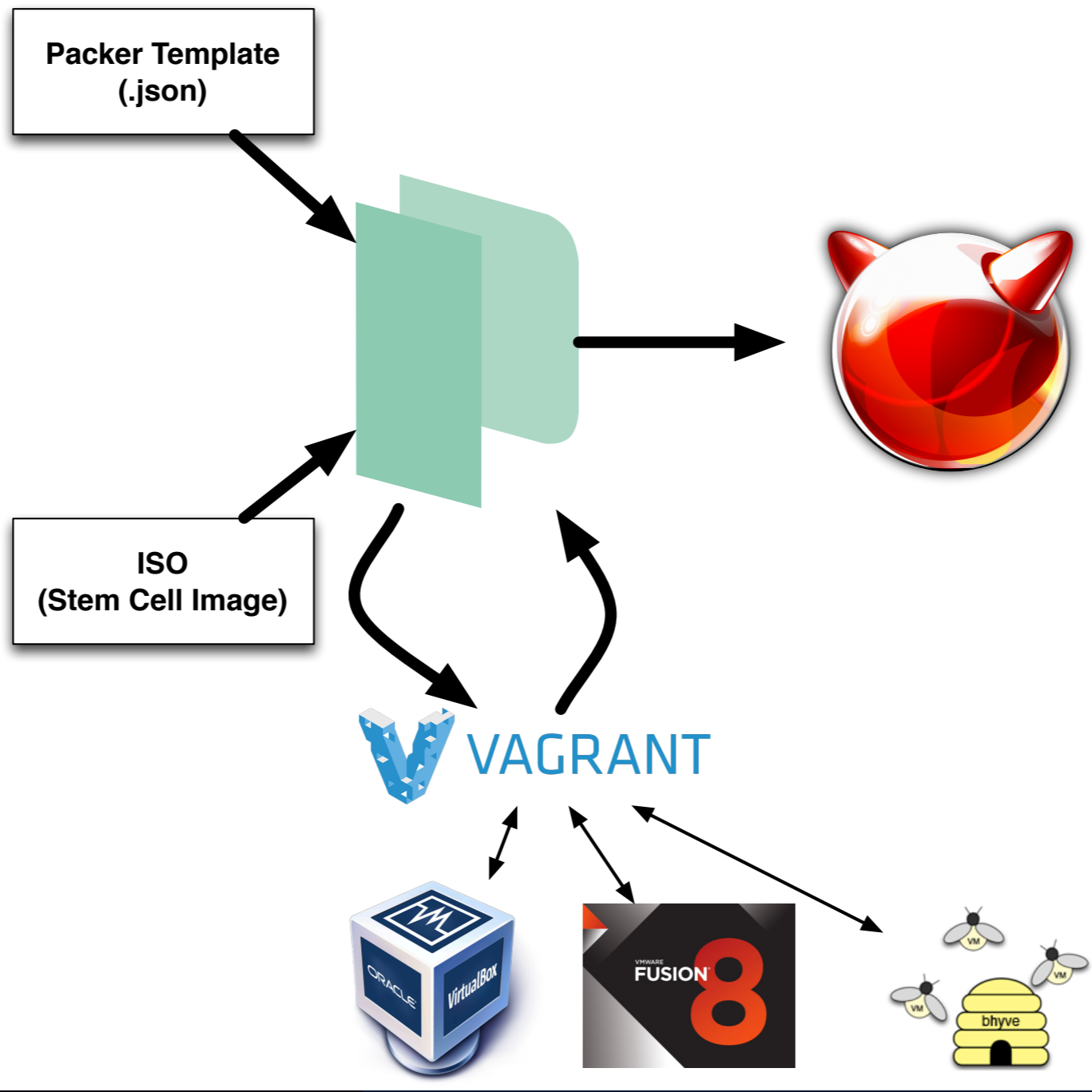
# Sean Chittenden

Engineering, HashiCorp

[sean@hashicorp.com](mailto:sean@hashicorp.com)

[seanc@FreeBSD.org](mailto:seanc@FreeBSD.org) (less retired)

# Quick Demo



# Awkward Little Chat



freeBSD

# Problem Child

# Problems

- Stable
- Debuggable
- Knowledgable Administrators
- Performance
- Secure
- Manageable
- Embeddable
- Pride



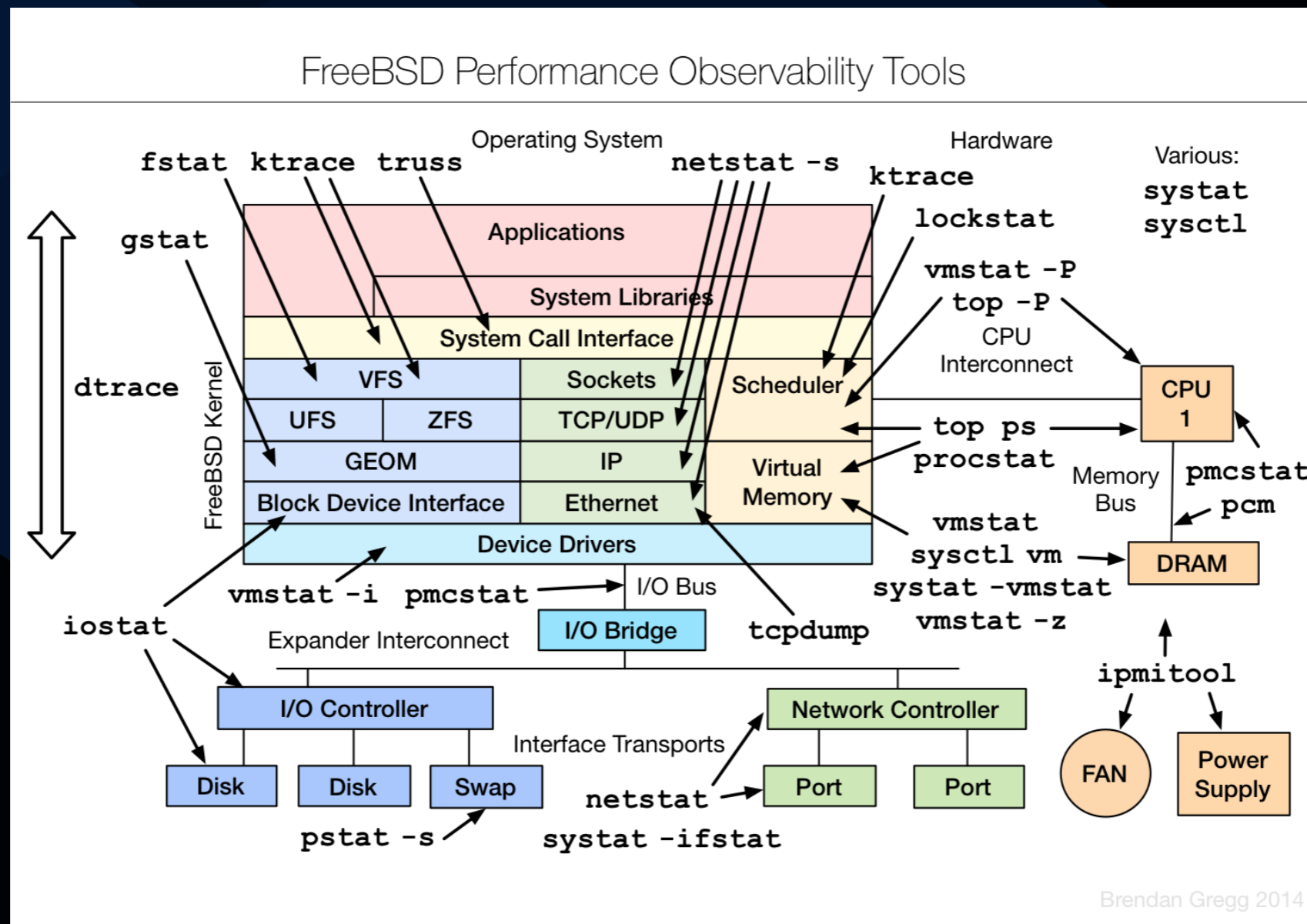
# Problem: Stability

# Problem: Stable

- Affinity for uptime
- Time required to install amortized over life of the server
- Artisanal configuration acceptable
- 900d uptimes are bragged about

# Problem: Debuggable

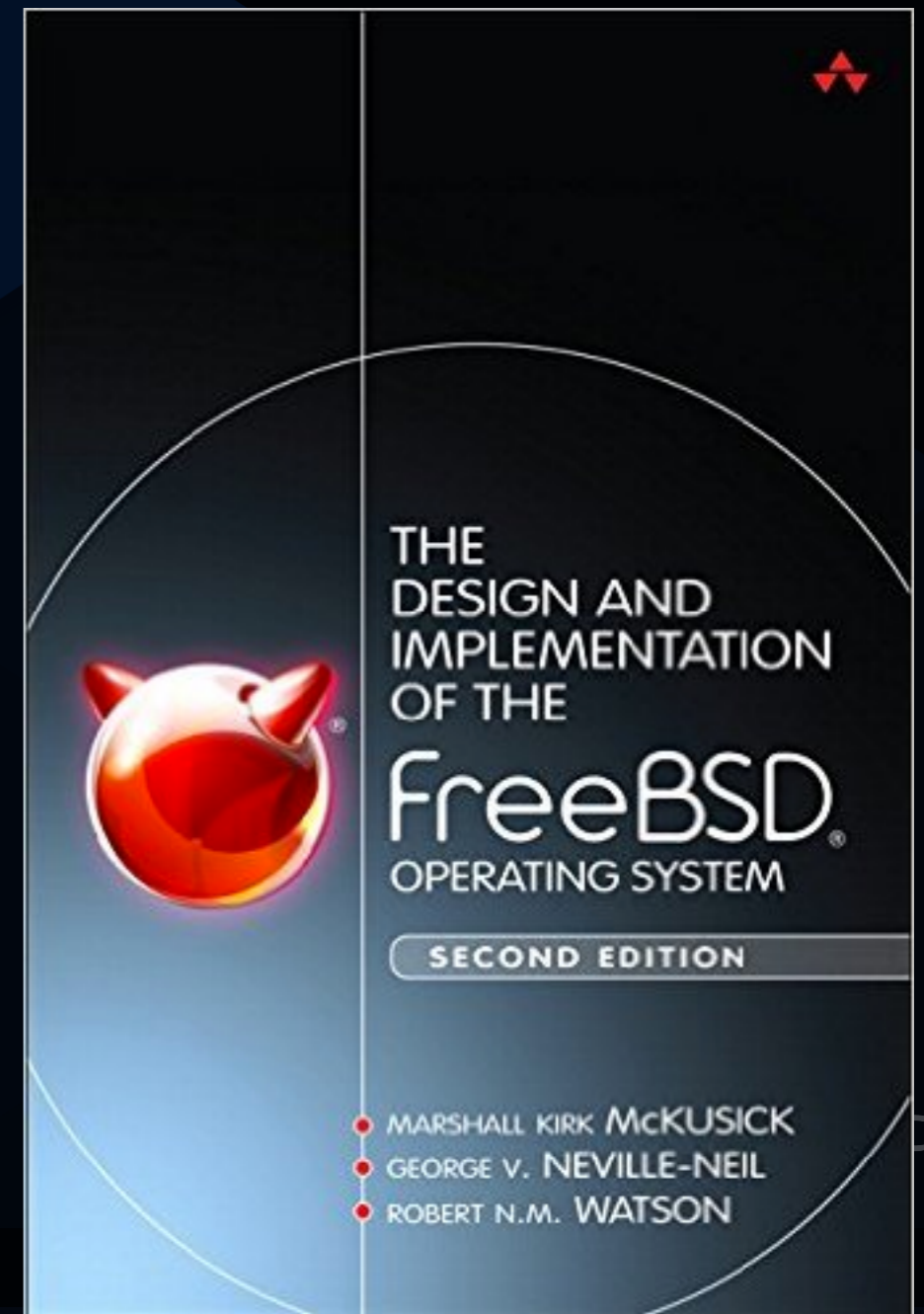
# Problem: Debuggable



# Problem: Knowledgeable Administrators

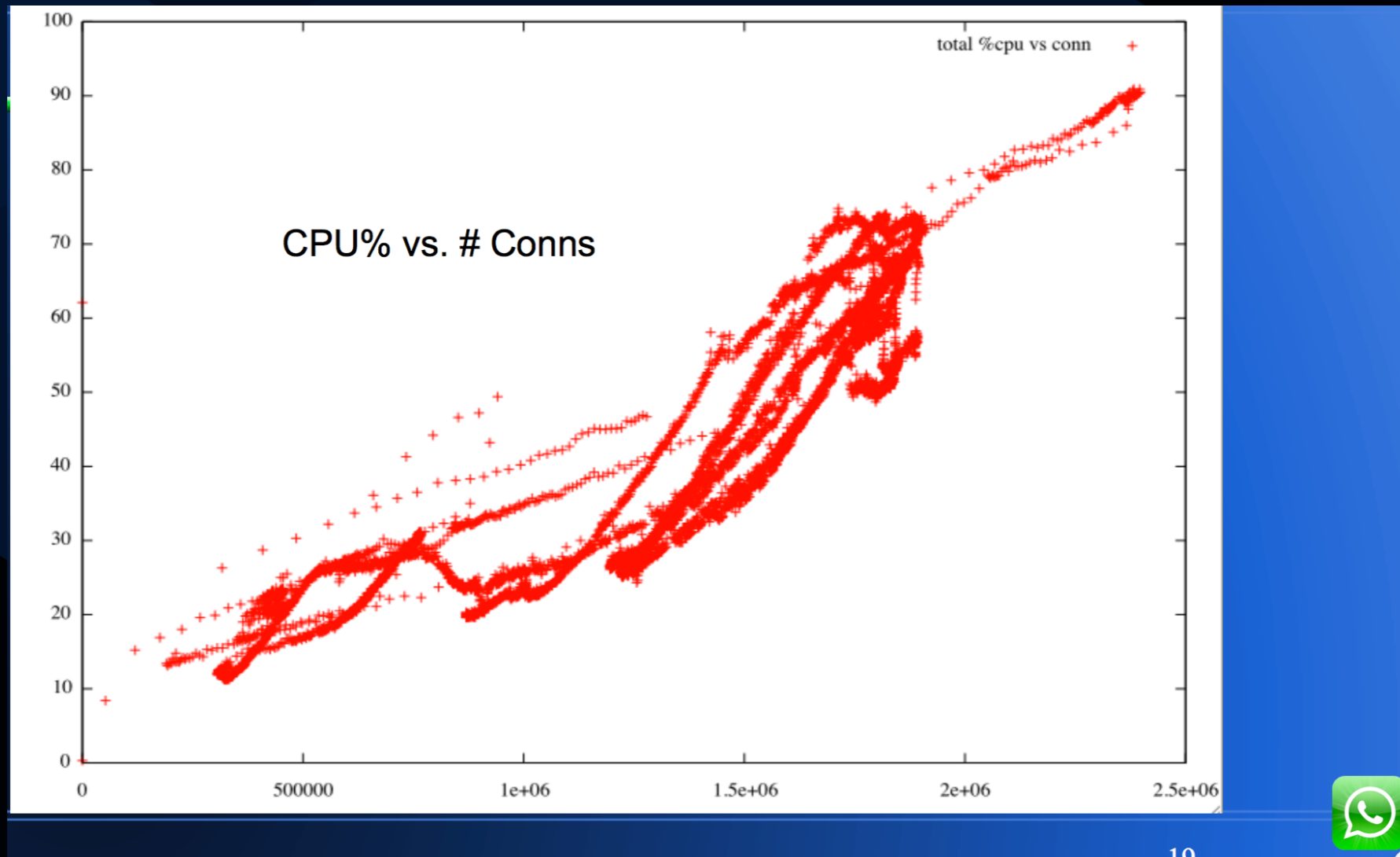
# Problems: Knowledgable Admins

- Comments
- Commit Comment Quality
- Documentation Project
  - Handbook and man(1) pages are excellent
- Lack of questions on Stack Overflow or other Search Engine-index sites



# Problem: Performance

# Problems: Performance



10



# Problems: Performance

## WhatsApp Blog



### 1 million is so 2011

Happy 2012 everyone!

A few months ago we published a blog post that talked about our servers doing 1 million tcp connections on a single box: <http://blog.whatsapp.com/?p=170>

Today we have an update for those keeping score at home: we are now able to easily push our systems to over 2 million tcp connections!

```
jkb@c123$ sysctl kern.ipc.numopensockets  
kern.ipc.numopensockets: 2277845
```

<https://blog.whatsapp.com/196/1-million-is-so-2011?>

# Problems: Performance

- ENOMEM
- `malloc(3)` can return `NULL`
- Fixed kernel/userland memory limits

# Problem: Secure

# Problems: Secure

- Deny by Default runs deep
- Network Services: OpenSSH, IPsec
- Firewall(s!) - `ipfw/pf`
- Yarrow (?)
- ACLs (discretionary enforcement)
- MAC (mandatory enforcement)

# Problem: Manageable

# Problems: Manageable

- Ports Tree
  - `pkg(1)` binary and src
  - `poudriere(1)`
- `make release`
- `mergemaster(1) -> etcupdate(1)`
- Packable Base / `{make,src}.conf`

# Problem: Embeddable

# Problems: Embeddable

- Appliance Vendors: stable, long lived branching
- Userland not exposed to customers
- Stripped down Kernel "just works, forever"
- Product life-cycles measured in years  
(vs software which is measured in months)



# Problem: Pride

# Problems: Pride

noun \ˈprīd\:

1. a high or inordinate opinion of one's own dignity, importance, merit, or superiority, whether as cherished in the mind or as displayed in bearing, conduct, etc.
3. a becoming or dignified sense of what is due to oneself or one's position or character; self-respect; self-esteem.
4. pleasure or satisfaction taken in something done by or belonging to oneself or believed to reflect credit upon oneself: civic pride.

<http://www.dictionary.com/browse/pride>

# Problems: Pride

noun, van·i·ty \ˈva-nə-tē\:

- the quality of people who have too much pride in their own appearance, abilities, achievements, etc. : the quality of being vain
- something (such as a belief or a way of behaving) which shows that you have too much pride in yourself, your social status, etc.

"Vanity - definitely my favorite sin." -Satan

<https://www.youtube.com/watch?v=Cv9zXUd55Sw>

<http://www.merriam-webster.com/dictionary/vanity>



Why are these problems?

# Wrong set of KPIs for Users

# Problems: KPIs

- Stable
- Debuggable
- Knowledgable Administrators
- Performance
- Secure
- Manageable
- Embeddable
- Pride

# Problems: KPIs

- Stable = Maximized Mean Time Between "Incident"
- Debuggable = Number of Unknowns
- Knowledgable Administrators = Number of servers per administrator

# Problems: KPIs

- Performance = Maximal number of Bits Chucked Per Server
- Performance = Bytes read/written per Server
- Secure = Mean Time Between Security Incident
- Secure = Average Severity of Security Incidents



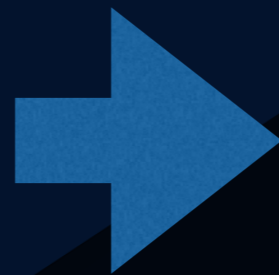
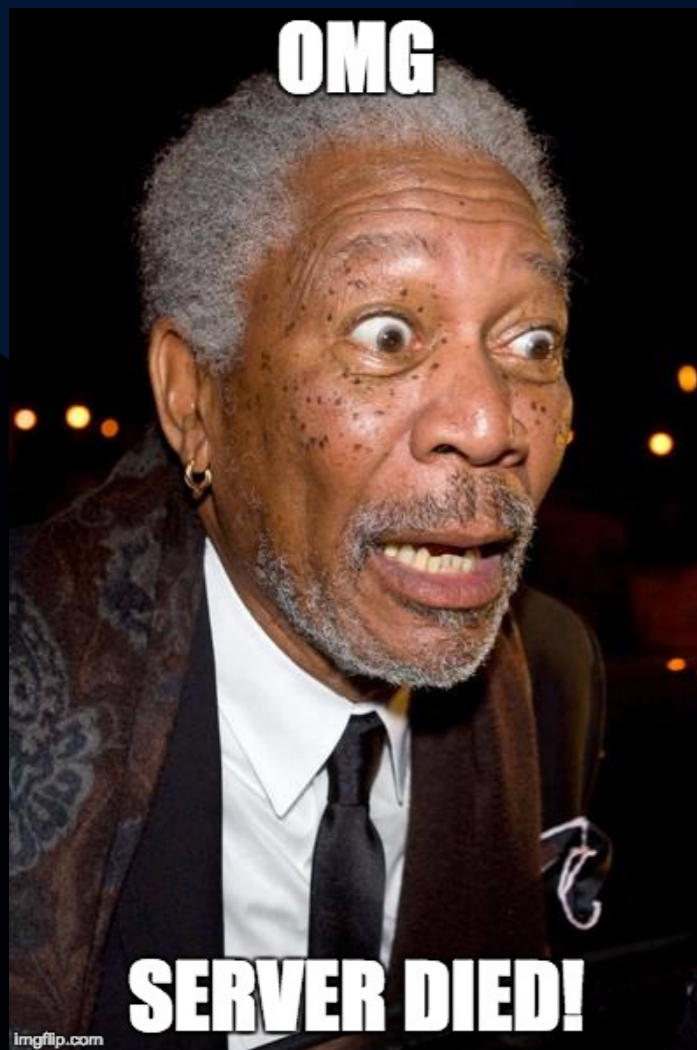
# Problems: KPIs

- Manageable = Mean Time Between Fresh Installs (MTTBFI?)
- Manageable = Maximize ROI per Server
- Embeddable = Maximize the number of random corners of the physical nether-verse BSD can be installed (small footprint and control)

# Problems: KPIs

Pride = Emotional investment or impact per server installed in production.

# Problems: KPIs



# Cloud KPIs for FreeBSD

# Cloud KPIs

- Friction: Effort required to spin up a new instance
- Street cred: Number of blog posts/Stack Overflow questions referring about  $\{TOPIC\}$  on  $\{AWS, GCP, DigitalOcean, etc\}$

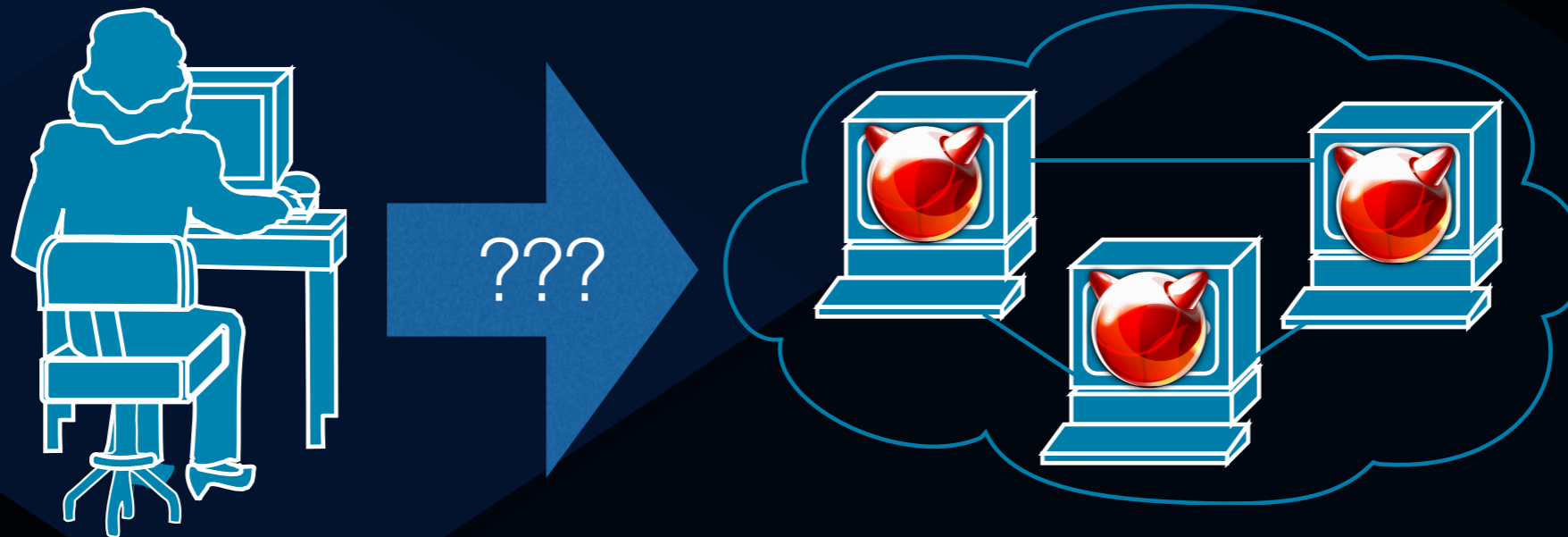
# Cloud KPI: Friction

- Number of steps required to create a new image
- Maximize Reliability of the build process for creating a new golden image
- Minimal number of differences between development and production

# Meatspace KPI: Distance



# Cloud KPI: Friction





# Modern Workflow

# Workflow

1. Spin up a VM
  - a. Dev
  - b. Test
  - c. Ship
2. Create golden image per app
3. Upload golden image
4. Scale golden image in production



VAGRANT

# Vagrant: Config (1/3)

```
% cat Vagrantfile
VAGRANTFILE_API_VERSION = "2"
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.guest = :freebsd
  config.vm.box = "freebsd/FreeBSD-11.0-CURRENT"
  config.ssh.shell = "sh"
  config.vm.synced_folder ".", "/vagrant", nfs: true, id: "vagrant-
root"

  # Insert Provider Snippet here
end
% cat Vagrantfile | wc -l
26
```

# Vagrant: Config VMware (2/3)

```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |
config|
  # Insert VM configs from 1/3 here

  config.vm.provider :vmware_fusion do |v|
    v.vmx["memsize"] = "4096"
    v.vmx["numvcpus"] = "8"
  end
end
```

# Vagrant: Config VirtualBox (3/3)

```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  # Insert VM configs from 1/3 here
  config.vm.provider :virtualbox do |vb|
    vb.gui = false
    vb.memory = "4096"
    vb.cpus = 8
    vb.customize ["modifyvm", :id, "--ioapic", "on"]
    vb.customize ["modifyvm", :id, "--hwvirtex", "on"]
    vb.customize ["modifyvm", :id, "--usb", "off"]
    vb.customize ["modifyvm", :id, "--usbhci", "off"]
    vb.customize ["modifyvm", :id, "--audio", "none"]
    vb.customize ["modifyvm", :id, "--nictype1", "virtio"]
    vb.customize ["modifyvm", :id, "--nictype2", "virtio"]
  end
end
```

# \$ vagrant up

```
$ vagrant up
```

```
Bringing machine 'default' up with 'vmware_fusion' provider...
```

```
==> default: Cloning VMware VM: 'freebsd/FreeBSD-11.0-CURRENT'. This can take some time...
```

```
==> default: Checking if box 'freebsd/FreeBSD-11.0-CURRENT' is up to date...
```

```
==> default: A newer version of the box 'freebsd/FreeBSD-11.0-CURRENT' is available! You currently
```

```
==> default: have version '2016.03.08'. The latest is version '2016.04.30'. Run
```

```
==> default: `vagrant box update` to update.
```

```
==> default: Verifying vmnet devices are healthy...
```

```
==> default: Preparing network adapters...
```

```
==> default: Starting the VMware VM...
```

```
==> default: Waiting for machine to boot. This may take a few minutes...
```

```
default: SSH address: 172.16.139.158:22
```

```
default: SSH username: vagrant
```

```
default: SSH auth method: private key
```

# \$ vagrant up

```
$ vagrant help up
```

```
Usage: vagrant up [options] [name]
```

```
Options:
```

```
--[no-]provision
```

```
Enable or disable provisioning
```

```
--provision-with x,y,z
```

```
Enable only certain provisioners, by type.
```

```
--[no-]destroy-on-error  
(default to true)
```

```
Destroy machine if any fatal error happens
```

```
--[no-]parallel  
supports it
```

```
Enable or disable parallelism if provider
```

```
--provider PROVIDER
```

```
Back the machine with a specific provider
```

```
-h, --help
```

```
Print this help
```



```
$ vagrant up
```

```
$ vagrant up --provider=vmware_fusion  
or
```

```
$ vagrant up --provider=virtualbox
```

```
...
```

# \$ vagrant up

```
$ vagrant up
```

```
[snip]
```

```
default: Warning: Connection refused. Retrying...
```

```
default: Warning: Connection refused. Retrying...
```

```
default: Warning: Remote connection disconnect. Retrying...
```

```
default: Warning: Connection refused. Retrying...
```

```
default: Warning: Connection timeout. Retrying...
```

```
default: Warning: Connection timeout. Retrying...
```

```
default: Warning: Connection refused. Retrying...
```

```
default:
```

```
default: Vagrant insecure key detected. Vagrant will automatically replace
```

```
default: this with a newly generated keypair for better security.
```

```
default:
```

```
default: Inserting generated public key within guest...
```

```
default: Removing insecure key from the guest if it's present...
```

# \$ vagrant up

```
$ vagrant up
```

```
[snip]
```

```
default: Key inserted! Disconnecting and reconnecting using new SSH key...
```

```
==> default: Machine booted and ready!
```

```
==> default: Forwarding ports...
```

```
default: -- 22 => 2222
```

```
==> default: Configuring network adapters within the VM...
```

```
The following SSH command responded with a non-zero exit status.
```

```
Vagrant assumes that this means the command failed!
```

```
sed -i '' -e '/^#VAGRANT-BEGIN/,/^#VAGRANT-END/ d' /etc/rc.conf
```

```
Stdout from the command:
```

# \$ vagrant up

```
$ vagrant up
```

```
[snip]
```

```
Stderr from the command:
```

```
sudo: error in /usr/local/etc/sudo.conf, line 0 while  
loading plugin `sudoers_policy'
```

```
sudo: unable to load /usr/local/libexec/sudo/sudoers.so:  
Shared object "libpam.so.6" not found, required by  
"sudoers.so"
```

```
sudo: fatal error, unable to load plugins
```

# \$ vagrant up

```
$ vagrant box update
```

```
==> default: Checking for updates to 'freebsd/FreeBSD-11.0-CURRENT'
```

```
default: Latest installed version: 2016.03.08
```

```
default: Version constraints:
```

```
default: Provider: vmware_desktop
```

```
==> default: Updating 'freebsd/FreeBSD-11.0-CURRENT' with provider 'vmware_desktop' from version
```

```
==> default: '2016.03.08' to '2016.04.30'...
```

```
==> default: Loading metadata for box 'https://atlas.hashicorp.com/freebsd/FreeBSD-11.0-CURRENT?access_token=62kMYhn6H0ZNBQ.atlasv1.XfHqcMrbfCd7HNFQoT7HVZ6rmICYI3lXx99nJr013yEJ1UsdxQGuXzE79ZFiqd04I4o'
```

```
==> default: Adding box 'freebsd/FreeBSD-11.0-CURRENT' (v2016.04.30) for provider: vmware_desktop
```

```
default: Downloading: https://atlas.hashicorp.com/freebsd/boxes/FreeBSD-11.0-CURRENT/versions/2016.04.30/providers/vmware\_desktop.box
```

```
[snip]
```

```
default: Downloading: https://atlas.hashicorp.com/freebsd/boxes/FreeBSD-11.0-CURRENT/versions/2016.04.30/providers/vmware\_desktop.box
```

```
==> default: Successfully added box 'freebsd/FreeBSD-11.0-CURRENT' (v2016.04.30) for 'vmware_desktop'!
```

# \$ vagrant up

```
$ vagrant up
```

```
Bringing machine 'default' up with 'vmware_fusion' provider...
```

```
==> default: Checking if box 'freebsd/FreeBSD-11.0-CURRENT' is up to date...
```

```
==> default: Machine is already running.
```

```
$ vagrant status
```

```
Current machine states:
```

```
default                running (vmware_fusion)
```

The VM is running. To stop this VM, you can run ``vagrant halt`` to shut it down, or you can run ``vagrant suspend`` to simply suspend the virtual machine. In either case, to restart it again, run ``vagrant up``.

```
$ vagrant ssh
```

```
$ vagrant ssh
```

```
FreeBSD 11.0-CURRENT (GENERIC) #0 r296485: Tue Mar 8 07:04:36  
UTC 2016
```

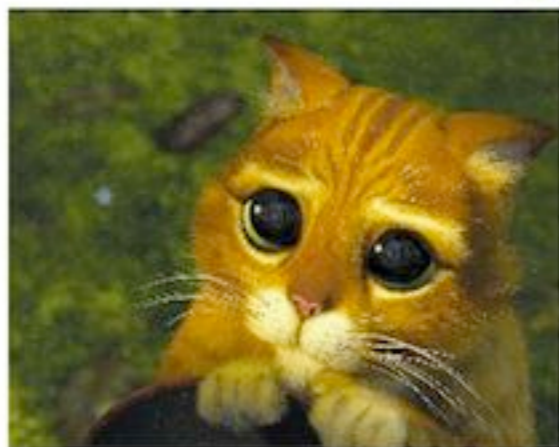
```
Welcome to FreeBSD!
```

```
[snip]
```

```
vagrant@:~ % uname -a
```

```
FreeBSD 11.0-CURRENT FreeBSD 11.0-CURRENT #0 r296485: Tue Mar  
8 07:04:36 UTC 2016 root@releng2.nyi.freebsd.org:/usr/obj/  
usr/src/sys/GENERIC amd64
```

## Service Model



- Pets are given names like pussinboots.cern.ch
- They are unique, lovingly hand raised and cared for
- When they get ill, you nurse them back to health



- Cattle are given numbers like vm0042.cern.ch
- They are almost identical to other cattle
- When they get ill, you get another one

- Future application architectures should use Cattle but Pets with strong configuration management are viable and still needed

<http://www.slideshare.net/gmccance/cern-data-centre-evolution>



# \$ vagrant destroy

```
$ vagrant destroy
```

```
    default: Are you sure you want to destroy the 'default'  
VM? [y/N] y
```

```
==> default: Stopping the VMware VM...
```

```
Connection to 172.16.139.158 closed by remote host.
```

```
==> default: Deleting the VM...
```

```
$
```

# \$ vagrant up

```
$ vagrant up
```

```
Bringing machine 'default' up with 'vmware_fusion' provider...
```

```
==> default: Cloning VMware VM: 'freebsd/FreeBSD-11.0-CURRENT'. This can take some time...
```

```
==> default: Checking if box 'freebsd/FreeBSD-11.0-CURRENT' is up to date...
```

```
==> default: Verifying vmnet devices are healthy...
```

```
==> default: Preparing network adapters...
```

```
==> default: Starting the VMware VM...
```

```
==> default: Waiting for machine to boot. This may take a few minutes...
```

```
default: SSH address: 172.16.139.159:22
```

```
default: SSH username: vagrant
```

```
default: SSH auth method: private key
```

```
default: Warning: Connection refused. Retrying...
```

```
[snip]
```

# \$ vagrant up

```
$ vagrant up
```

```
[snip]
```

```
==> default: Configuring network adapters within the VM...
```

```
The following SSH command responded with a non-zero exit status.
```

```
Vagrant assumes that this means the command failed!
```

```
sed -i '' -e '/^#VAGRANT-BEGIN/,/^#VAGRANT-END/ d' /etc/rc.conf
```

```
Stdout from the command:
```

```
Stderr from the command:
```

```
sudo: error in /usr/local/etc/sudo.conf, line 0 while loading plugin `sudoers_policy'
```

```
sudo: unable to load /usr/local/libexec/sudo/sudoers.so: Shared object "libpam.so.6" not found, required by "sudoers.so"
```

```
sudo: fatal error, unable to load plugins
```

# \$ vagrant up

```
$ vagrant ssh
```

```
$ vagrant@:~ % sudo tcsh
```

```
sudo: error in /usr/local/etc/sudo.conf, line 0 while loading plugin  
`sudoers_policy'
```

```
sudo: unable to load /usr/local/libexec/sudo/sudoers.so: Shared object "libpam.so.  
6" not found, required by "sudoers.so"
```

```
sudo: fatal error, unable to load plugins
```

```
vagrant@:~ % su
```

```
Password:
```

```
root@:/home/vagrant # pkg delete sudo
```

```
root@:/home/vagrant # pkg install -y sudo
```

```
$ vagrant reload
```

```
[blows up again]
```

# \$ vagrant up

```
vagrant@:~ % ldd /usr/local/libexec/sudo/sudoers.so
/usr/local/libexec/sudo/sudoers.so:
  libbsm.so.3 => /usr/lib/libbsm.so.3 (0x801246000)
  libutil.so.9 => /lib/libutil.so.9 (0x801461000)
  libpam.so.6 => not found (0)
  libsudo_util.so.0 => /usr/local/libexec/sudo/libsudo_util.so.0
(0x801674000)
  libintl.so.8 => /usr/local/lib/libintl.so.8 (0x801887000)
  libz.so.6 => /lib/libz.so.6 (0x801a91000)
  libc.so.7 => /lib/libc.so.7 (0x800823000)
```

# \$ vagrant up

```
vagrant@:~ % su
```

```
Password:
```

```
root@:~vagrant # cd /usr/lib
```

```
root@:/usr/lib # ln -s libpam.so.5 libpam.so.6
```

```
root@:/usr/lib # ldd /usr/local/libexec/sudo/sudoers.so
```

```
/usr/local/libexec/sudo/sudoers.so:
```

```
libbsm.so.3 => /usr/lib/libbsm.so.3 (0x801246000)
```

```
libutil.so.9 => /lib/libutil.so.9 (0x801461000)
```

```
libpam.so.6 => /usr/lib/libpam.so.6 (0x801674000)
```

```
libsudo_util.so.0 => /usr/local/libexec/sudo/libsudo_util.so.0 (0x801881000)
```

```
libintl.so.8 => /usr/local/lib/libintl.so.8 (0x801a94000)
```

```
libz.so.6 => /lib/libz.so.6 (0x801c9e000)
```

```
libc.so.7 => /lib/libc.so.7 (0x800823000)
```

# \$ vagrant reload

```
$ vagrant reload
==> default: Attempting graceful shutdown of VM...
Connection to 172.16.139.159 closed by remote host.
==> default: Checking if box 'freebsd/FreeBSD-11.0-CURRENT' is up to date...
==> default: Verifying vmnet devices are healthy...
==> default: Preparing network adapters...
==> default: Starting the VMware VM...
==> default: Waiting for machine to boot. This may take a few minutes...
  default: SSH address: 172.16.139.159:22
  default: SSH username: vagrant
  default: SSH auth method: private key
  default: Warning: Connection timeout. Retrying...
  default: Warning: Host appears down. Retrying...
  default: Warning: Connection refused. Retrying...
==> default: Machine booted and ready!
==> default: Forwarding ports...
  default: -- 22 => 2222
==> default: Configuring network adapters within the VM...
==> default: Exporting NFS shared folders...
==> default: Preparing to edit /etc/exports. Administrator privileges will be required...
==> default: Mounting NFS shared folders...
```

# Make Development Great Again

```
my-laptop$ ls -lA
total 8
drwxr-xr-x  3 sean  staff  102 Jun 11 13:12 .vagrant
-rw-r--r--  1 sean  staff  867 Jun 11 13:11 Vagrantfile
my-laptop$ vagrant ssh
vagrant@:~ % ll /vagrant/
total 8
drwxr-xr-x  3 501  staff  102 Jun 11 17:12 .vagrant/
-rw-r--r--  1 501  staff  867 Jun 11 17:11 Vagrantfile
vagrant@:~ % mount
/dev/gpt/rootfs on / (ufs, local, soft-updates)
devfs on /dev (devfs, local, multilabel)
172.16.139.1:/Users/sean/src/FreeBSD/vagrant/vm-nfs on /vagrant (nfs)
```



# \$ vagrant suspend

```
vagrant@:~ % echo gozfraba > /vagrant/foo
```

```
vagrant@:~ % logout
```

```
Shared connection to 172.16.139.159 closed.
```

```
my-laptop $ cat foo
```

```
gozfraba
```

```
my-laptop $ vagrant suspend
```

```
==> default: Suspending the VMware VM...
```

```
my-laptop $ vagrant status
```

```
Current machine states:
```

```
default                suspended (vmware_fusion)
```

The VM is suspended. To resume this VM, run `vagrant up`.

# Scripted Hotness

```
% cat Vagrantfile
[snip]
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|

  config.vm.guest = :freebsd

  config.vm.box = "freebsd/FreeBSD-11.0-CURRENT"

  config.ssh.shell = "sh"

  config.vm.synced_folder ".", "/vagrant", nfs: true, id:
"vagrant-root"

config.vm.provision "shell", inline: $script, privileged: false

[snip]
```

# Scripted Hotness

```
% cat Vagrantfile
$script = <<SCRIPT

sudo pkg install -y go runit

SCRIPT

[snip]
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|

  config.vm.guest = :freebsd

  config.vm.box = "freebsd/FreeBSD-11.0-CURRENT"

  config.ssh.shell = "sh"

  config.vm.synced_folder ".", "/vagrant", nfs: true, id: "vagrant-root"

  config.vm.provision "shell", inline: $script, privileged: false

[snip]
```

# vagrant + bhyve?

jesa7955 / vagrant-bhyve

Unwatch 3

Unstar 2

Fork 0

Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

No description or website provided.

11 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload new file

Find file

Clone or download

jesa7955 add load\_os action and its corresponding driver code

Latest commit 1f... 7 hours ago

# vagrant + bhyve?

<https://github.com/jesa7955/vagrant-bhyve>

```
{  
  "provider"      : "libvirt",  
  "loader"        : "bhyveload"  
}
```

```
$ vagrant up
```

```
$ vagrant up --provider=vmware_fusion  
or
```

```
$ vagrant up --provider=virtualbox  
or
```

```
$ vagrant up --provider=bhyve
```

# Serverless Clusters

```
$ cat Vagrantfile
```

```
[snip]
```

```
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|  
  1.upto(3) do |n|  
    vmName = "nomad-server%02d" % [n]  
    config.vm.define vmName, autostart: (n == 1 ? true : false), primary: (n == 1 ? true : false) do |vmCfg|  
      vmCfg.vm.hostname = vmName  
      vmCfg = configureVM(vmCfg)  
    end  
  end  
  
  1.upto(3) do |n|  
    vmName = "nomad-client%02d" % [n]  
    config.vm.define vmName, autostart: false, primary: false do |vmCfg|  
      vmCfg.vm.hostname = vmName  
      vmCfg = configureVM(vmCfg)  
    end  
  end  
end
```

# Serverless Clusters

```
# Launch nomad-server01
```

```
$ vagrant up
```

```
# Suspends nomad-server01
```

```
$ vagrant suspend
```

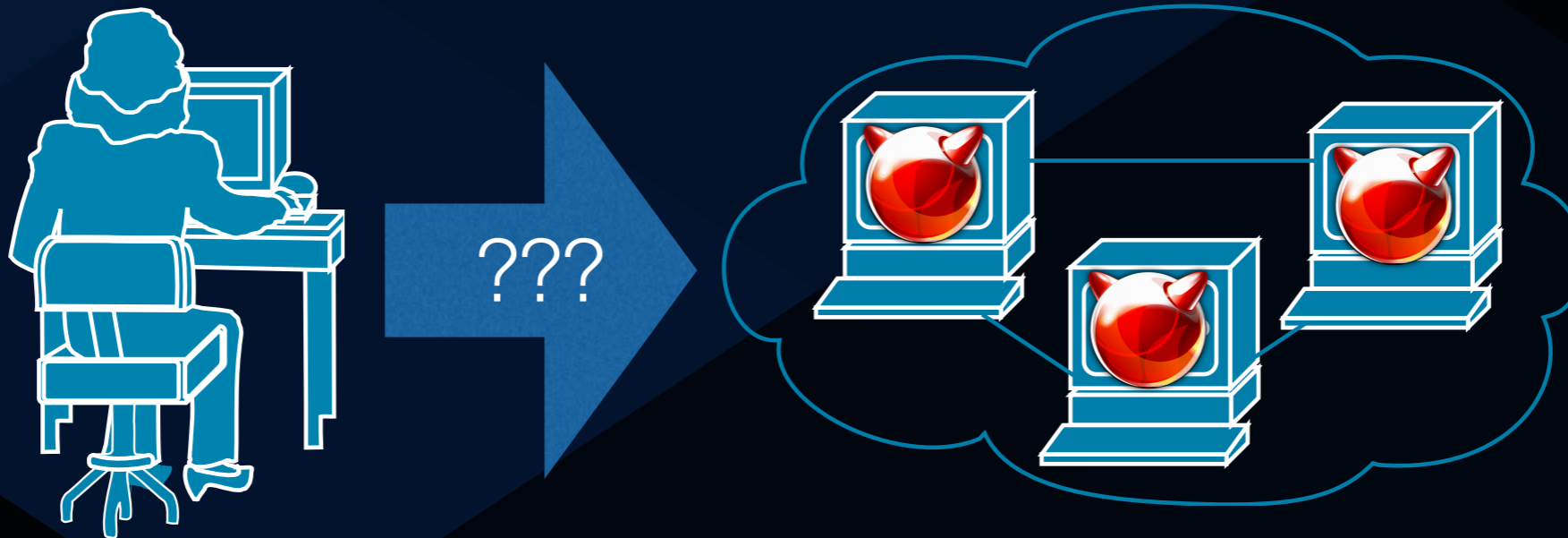
```
$ vagrant up '/nomad-server*/' nomad-client01
```





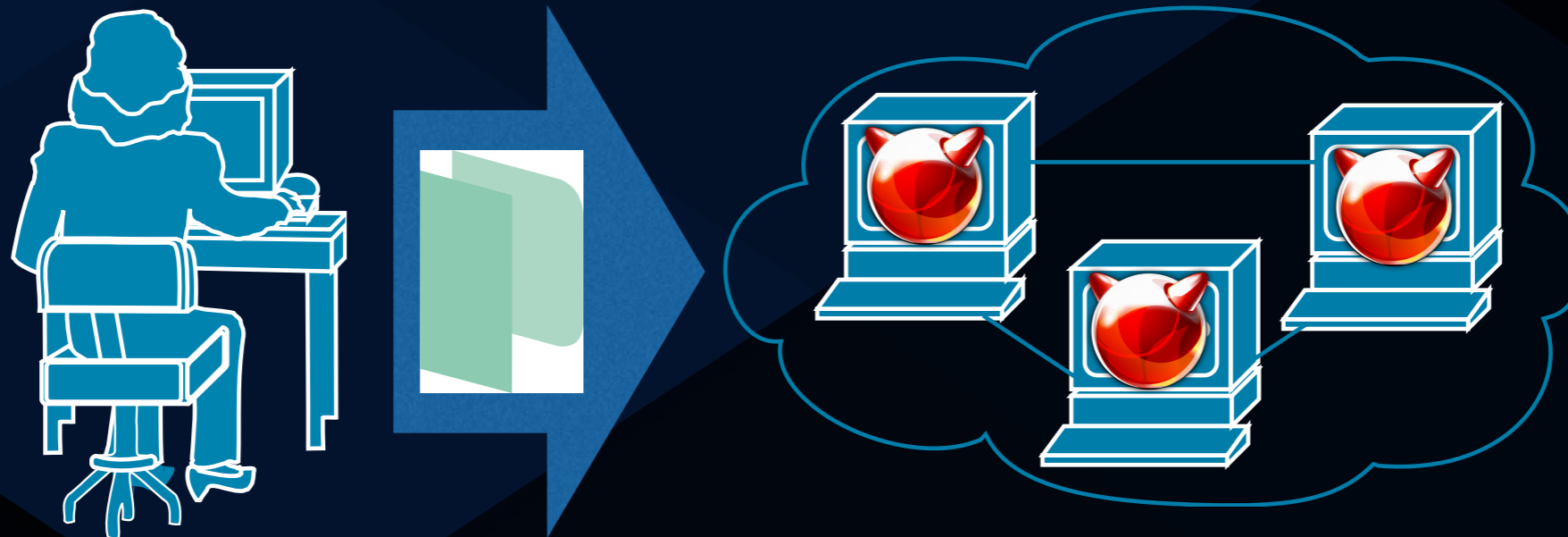
# Golden Image

# Dev to Prod?

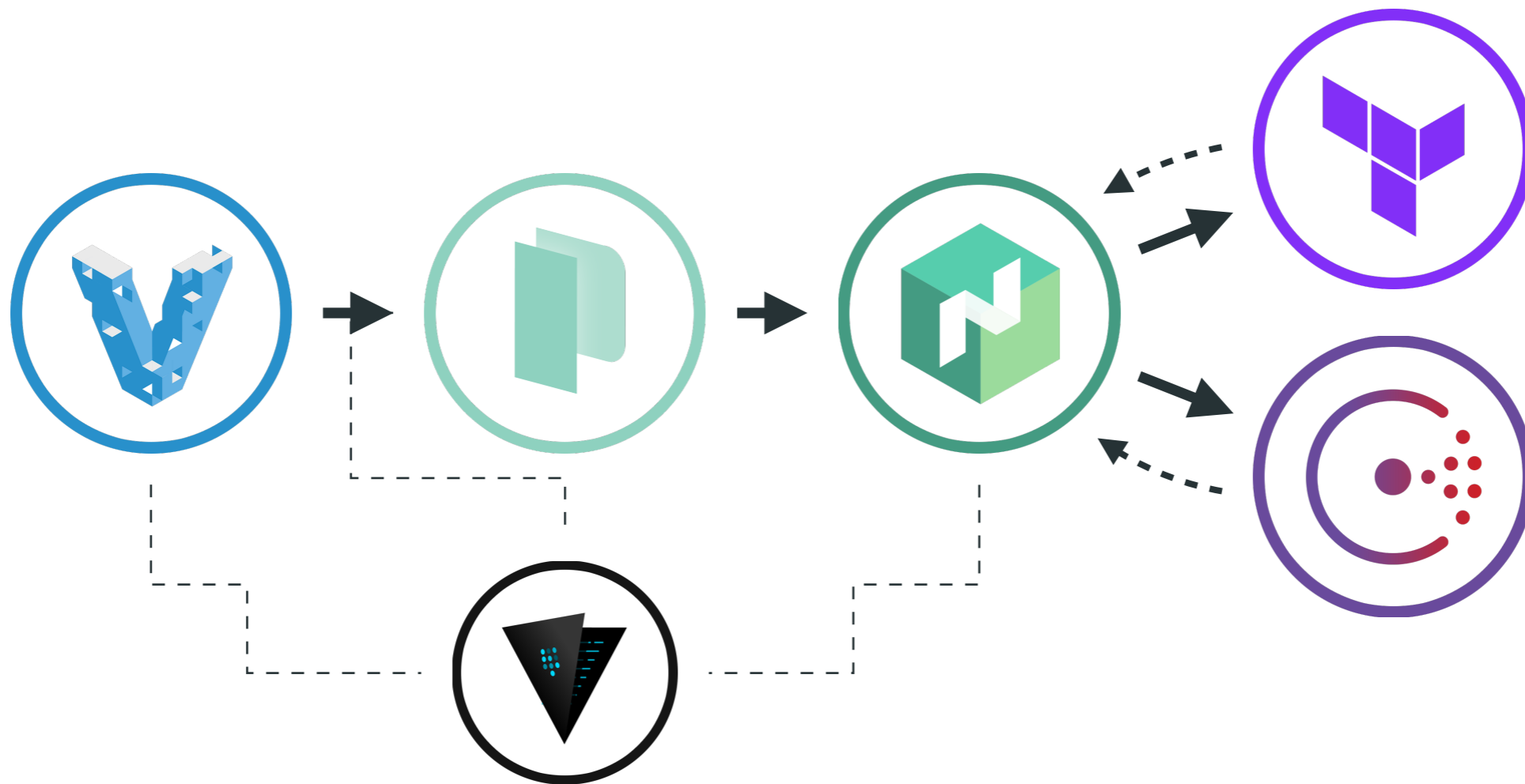




# Packer FTW



# (Part of a Larger Ecosystem)



# Packer Templates

```
# Hat tip to brd@ for doing the initial heavy lifting!  
$ git clone https://github.com/brd/packer-freebsd.git  
  
$ cd packer-freebsd  
  
$ ./automatic-11.0-current-ufs.sh --only=vmware-iso  
[snip]  
  
$ vagrant up  
  
$ vagrant ssh
```

# Packer Templates

```
$ ./automatic-11.0-current-ufs.sh --only=vmware-iso
```

```
Looking for latest -CURRENT from ftp://ftp.FreeBSD.org/pub/FreeBSD/snapshots/ISO-IMAGES/11.0/ ...
```

```
Using FreeBSD-11.0-CURRENT-amd64-20160518-r300097-disc1.iso (checksum  
8b5c9a9240962a497507f374dcf3839744c63f271fe2b2d36a4fd4d46eb966a3a697684f3379dcb87816b92b44f443e5fa4a71ec713fc408c  
1ac60990d73817e) ...
```

```
vmware-iso output will be in this color.
```

```
==> vmware-iso: Downloading or copying ISO
```

```
vmware-iso: Downloading or copying: http://ftp.freebsd.org/pub/FreeBSD/snapshots/ISO-IMAGES/11.0/  
FreeBSD-11.0-CURRENT-amd64-20160518-r300097-disc1.iso
```

```
==> vmware-iso: Creating virtual machine disk
```

```
==> vmware-iso: Building and writing VMX file
```

```
==> vmware-iso: Starting virtual machine...
```

```
vmware-iso: The VM will be run headless, without a GUI. If you want to
```

```
vmware-iso: view the screen of the VM, connect via VNC without a password to
```

```
vmware-iso: 127.0.0.1:5997
```

```
==> vmware-iso: Waiting 45s for boot...
```

```
==> vmware-iso: Connecting to VM via VNC
```

```
==> vmware-iso: Error connecting to VNC: dial tcp 127.0.0.1:5965: getsockopt: connection refused
```

```
==> vmware-iso: Typing the boot command over VNC...
```

# Packer Templates

```
$ ./automatic-11.0-current-ufs.sh --only=vmware-iso
```



# Packer Templates

```
$ ./automatic-11.0-current-ufs.sh --only=vmware-iso
```

```
Looking for latest -CURRENT from ftp://ftp.FreeBSD.org/pub/FreeBSD/snapshots/ISO-IMAGES/11.0/ ...
```

```
Using FreeBSD-11.0-CURRENT-amd64-20160518-r300097-disc1.iso (checksum  
8b5c9a9240962a497507f374dcf3839744c63f271fe2b2d36a4fd4d46eb966a3a697684f3379dcb87816b92b44f443e5fa4a71ec713fc408c  
1ac60990d73817e) ...
```

```
vmware-iso output will be in this color.
```

```
==> vmware-iso: Downloading or copying ISO
```

```
vmware-iso: Downloading or copying: http://ftp.freebsd.org/pub/FreeBSD/snapshots/ISO-IMAGES/11.0/  
FreeBSD-11.0-CURRENT-amd64-20160518-r300097-disc1.iso
```

```
==> vmware-iso: Creating virtual machine disk
```

```
==> vmware-iso: Building and writing VMX file
```

```
==> vmware-iso: Starting virtual machine...
```

```
vmware-iso: The VM will be run headless, without a GUI. If you want to
```

```
vmware-iso: view the screen of the VM, connect via VNC without a password to
```

```
vmware-iso: 127.0.0.1:5997
```

```
==> vmware-iso: Waiting 45s for boot...
```

```
==> vmware-iso: Connecting to VM via VNC
```

```
==> vmware-iso: Error connecting to VNC: dial tcp 127.0.0.1:5965: getsockopt: connection refused
```

```
==> vmware-iso: Typing the boot command over VNC...
```

Questions?

# Thanks!

[sean@hashicorp.com](mailto:sean@hashicorp.com)