

# **vmpooler**

pdxdevops : April 2015

# who am I?



Scott Schneider

Quality Engineer @ Puppet Labs

✉ [sschneider@puppetlabs.com](mailto:sschneider@puppetlabs.com)

🐙 [github.com/sschneid](https://github.com/sschneid)

🐦 [@lrfsfsh](https://twitter.com/lrfsfsh)

# what's a vmpooler?

instantly-available VMs

# what's a vmpooler?

## instantly-available VMs

```
$ time( curl -d --url vmpooler/vm/debian-7-x86_64 | ping -c1 `sed -n 's/"hostname": "\(.*\)/\1/p'` )
```

```
PING ra16cif18962djo.delivery.puppetlabs.net (10.32.120.134): 56 data bytes
```

```
64 bytes from 10.32.120.134: icmp_seq=0 ttl=62 time=18.763 ms
```

```
--- ra16cif18962djo.delivery.puppetlabs.net ping statistics ---
```

```
1 packets transmitted, 1 packets received, 0.0% packet loss
```

```
round-trip min/avg/max/stddev = 18.763/18.763/18.763/0.000 ms
```

```
real 0m0.142s
```

```
user 0m0.007s
```

```
sys 0m0.009s
```

# why?

ec2 is expensive, and provisioning sometimes fails

*we spent how much last month?!?*

*why did you only give me 42 VMs when I asked for 50?*

cloning on-demand (vSphere) isn't fast enough

*clone time (8s) + boot-up time (120s) = ZZZzzz...*

# why?

## user stories

*as an acceptance test*

*I don't want to bankrupt my employer*

*I don't want to deal with provisioning failures*

*I don't want to wait for a VM to test on*

# why?

## user stories

*as an acceptance test*

*I don't want to bankrupt my employer*

*I don't want to deal with provisioning failures*

*I don't want to wait for a VM to test on*

*as a developer*

*getting a host via a simple curl interface is faster and less cumbersome than finding the right box somewhere on the internet and using Vagrant to achieve the same thing*

# why?

## user stories

*as an acceptance test*

*I don't want to bankrupt my employer*

*I don't want to deal with provisioning failures*

*I don't want to wait for a VM to test on*

*as a developer*

~~*getting a host via a simple curl interface is faster and less cumbersome than finding the right box somewhere on the internet and using Vagrant to achieve the same thing*~~

*I don't want to manage Virtual Machine images*



# why?

## user stories

*as an acceptance test*

*I don't want to bankrupt my employer*

*I don't want to deal with provisioning failures*

*I don't want to wait for a VM to test on*

*as a developer*

~~*getting a host via a simple curl interface is faster and less cumbersome than finding the right box somewhere on the internet and using Vagrant to achieve the same thing*~~

*I don't want to manage Virtual Machine images*

*especially 74 (!) of them*

# how?

## 1) define “pools” in a YAML config

```
- name: 'debian-7-i386'  
  template: 'Templates/debian-7-i386'  
  folder: 'Pooled VMs/debian-7-i386'  
  pool: 'Pooled VMs/debian-7-i386'  
  datastore: 'vmstorage'  
  size: 5
```

# how?

## 1) define “pools” in a YAML config

```
- name: 'debian-7-i386'  
  template: 'Templates/debian-7-i386'  
  folder: 'Pooled VMs/debian-7-i386'  
  pool: 'Pooled VMs/debian-7-i386'  
  datastore: 'vmstorage'  
  size: 5
```

## 2) vmpooler fills up VM pools to specified size

```
[2015-04-15 11:54:23] [ ] [debian-7-x86_64] '119qp943awg4lqz' is being cloned from 'debian-7-x86_64'  
[2015-04-15 11:54:29] [+] [debian-7-x86_64] '119qp943awg4lqz' cloned from 'debian-7-x86_64' in 5.88 seconds  
[2015-04-15 11:56:14] [>] [debian-7-x86_64] '119qp943awg4lqz' moved to 'ready' queue
```

# how?

## 1) define “pools” in a YAML config

```
- name: 'debian-7-i386'  
  template: 'Templates/debian-7-i386'  
  folder: 'Pooled VMs/debian-7-i386'  
  pool: 'Pooled VMs/debian-7-i386'  
  datastore: 'vmstorage'  
  size: 5
```

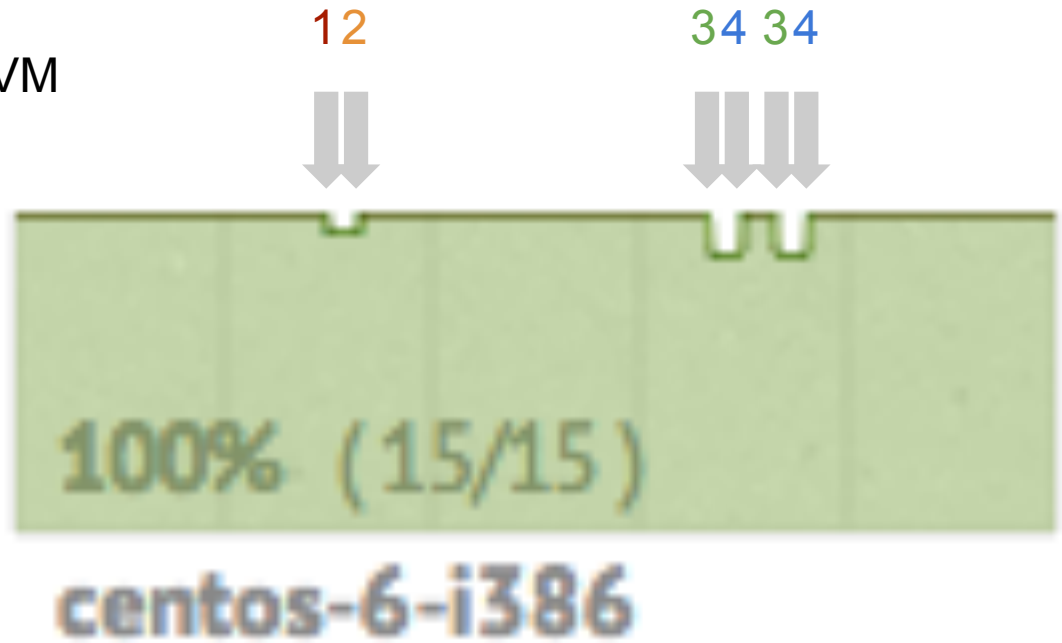
## 2) vmpooler fills up VM pools to specified size

```
[2015-04-15 11:54:23] [ ] [debian-7-x86_64] '119qp943awg4lqz' is being cloned from 'debian-7-x86_64'  
[2015-04-15 11:54:29] [+] [debian-7-x86_64] '119qp943awg4lqz' cloned from 'debian-7-x86_64' in 5.88 seconds  
[2015-04-15 11:56:14] [>] [debian-7-x86_64] '119qp943awg4lqz' moved to 'ready' queue
```

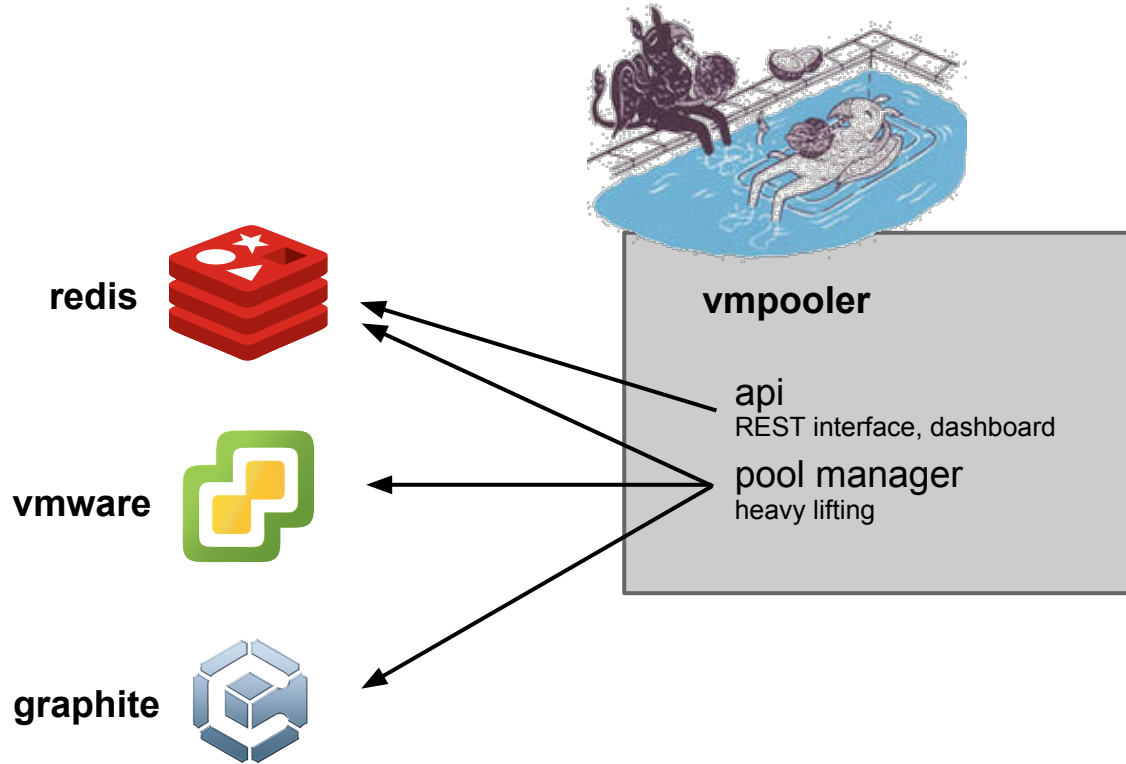
## 3) pools are back-filled to specified size when VMs are checked-out

# how?

- 1) client requests 1 VM
- 2) vmpooler sees pool at 14/15, provisions and registers new VM
- 3) client requests 2 VMs
- 4) pool at 13/15, better add 2 more VMs...











# architecture



# isn't it a waste of resources though?

not really

*waiting VMs are idle, don't really use up host resources*

Name	1 ▲	Status	Host CPU	Host Mem	Guest Mem - %
 a6yhht3de984slv		✓ Normal	0 MHz	389 MB	0 
 aa90drmjxxjewmp		✓ Normal	0 MHz	381 MB	0 
 aqv6ofthhazkvqr		✓ Normal	0 MHz	397 MB	0 
 arq85yuohzw79d3		✓ Normal	0 MHz	411 MB	0 

# will my infrastructure catch on fire?

vmpooler only provisions to “green” hosts

*not hosts with warnings (high CPU, high memory)*

*not hosts in maintenance mode*

auto-balances across ESXi hosts

*no DRS required!*

configurable VM auto-reaping (TTLs)

*vm\_lifetime: 12*



# what else does it do?

REST interface

*/api/v1*

metrics

*everyone loves a metric!*

dashboard

*looks delightful on a flatscreen*

# what else does it do?

## REST interface

*/api/v1*

## metrics

*everyone loves a metric!*

## dashboard

*looks delightful on a flatscreen*

# what else does it do? : REST interface

```
$ curl --url vmpooler.company.com/vm
```

```
[  
  "debian-7-i386",  
  "debian-7-x86_64"  
]
```

```
$ curl -d '{"debian-7-i386":"2","debian-7-x86_64":"1"}' --url vmpooler.company.com/vm
```

```
{  
  "ok": true,  
  "debian-7-i386": {  
    "hostname": [  
      "o41xtodlvnvu5cw",  
      "khirruvwfjlmx3y"  
    ]  
  },  
  "debian-7-x86_64": {  
    "hostname": "y91qbrpbj6d13q"  
  }  
}
```

# what else does it do? : REST interface

```
$ curl --url vmpooler.company.com/vm/o41xtodlvnvu5cw
```

```
{  
  "ok": true,  
  "o41xtodlvnvu5cw": {  
    "template": "debian-7-i386",  
    "lifetime": 12,  
    "running": 3,  
    "domain": "company.com"  
  }  
}
```

```
$ curl -X DELETE --url vmpooler.company.com/vm/o41xtodlvnvu5cw
```

```
{  
  "ok": true  
}
```

# what else does it do?

REST interface

*/api/v1*

**metrics**

*everyone loves a metric!*

dashboard

*looks delightful on a flatscreen*

# what else does it do? : metrics

```
$ curl --url vmpooler.company.com/status
```

```
{  
  "capacity": {  
    "current": 716,  
    "total": 717,  
    "percent": 99.9  
  },  
  "clone": {  
    "duration": {  
      "average": 8.8,  
      "min": 2.79,  
      "max": 69.76  
    },  
    "count": {  
      "total": 1779  
    }  
  },  
  "queue": {  
    "pending": 1,  
    "cloning": 0,  
    "booting": 1,  
    "ready": 716,  
  }  
}
```

# what else does it do? : metrics

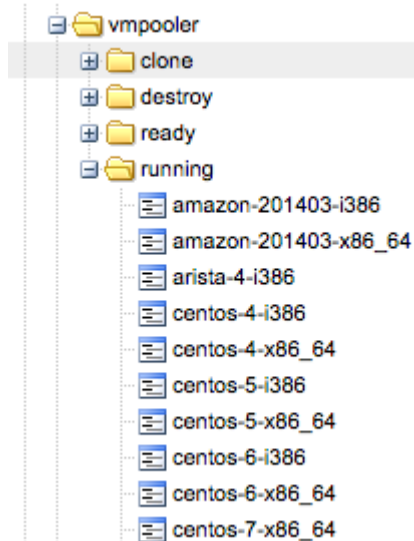
```
$ curl --url vmpooler.company.com/summary?from=2015-03-11
```

```
"daily": [  
  {  
    "date": "2015-03-11",  
    "boot": {  
      "duration": {  
        "average": 106.6,  
        "min": 83.09,  
        "max": 121.06,  
        "total": 639.36  
      },  
      "count": {  
        "total": 6  
      }  
    },  
    "clone": {  
      "duration": {  
        "average": 4.6,  
        "min": 2.78,  
        "max": 8.1,  
        "total": 63.94  
      }  
    }  
  }  
]
```

# what else does it do? : metrics

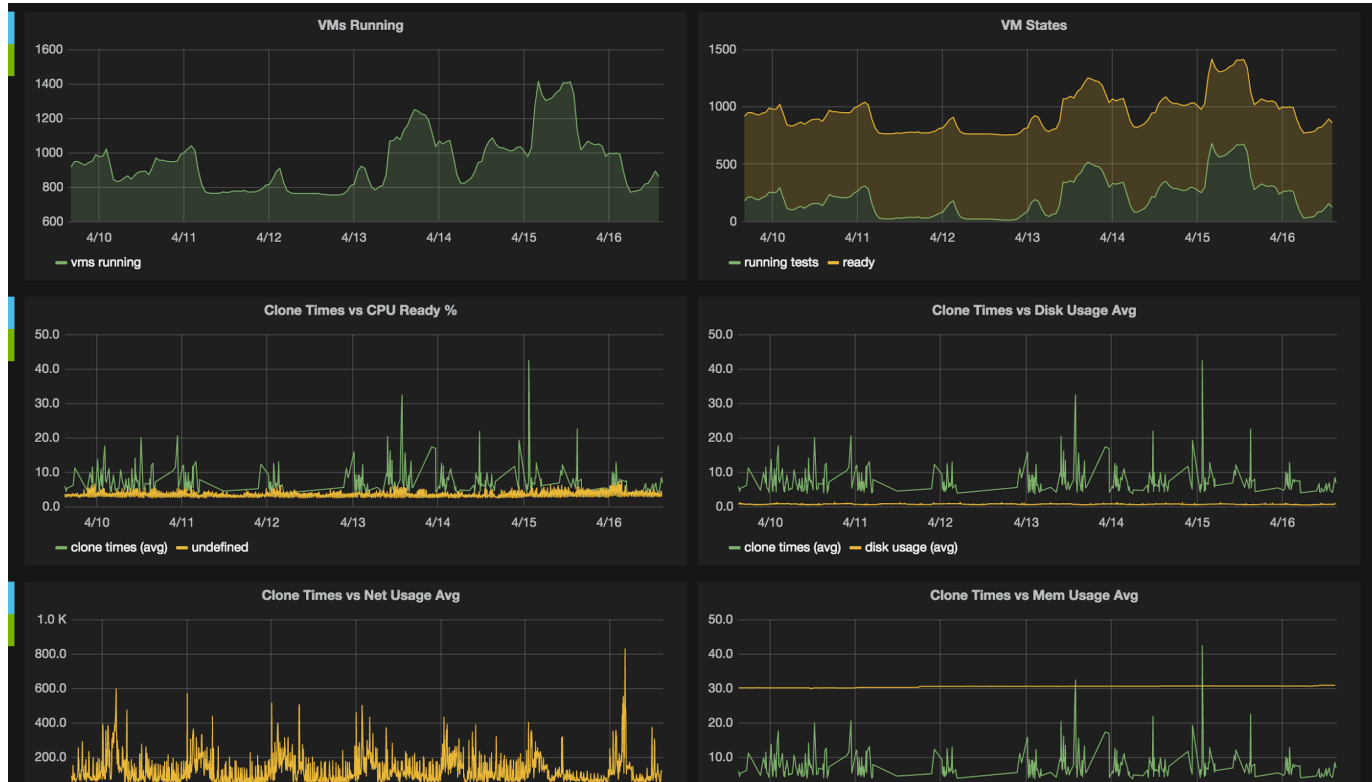
## Graphite

```
:graphite:  
  server: 'graphite.company.com'  
  prefix: 'vmpooler'
```





# what else does it do? : metrics



# what else does it do?

REST interface

*/api/v1*

metrics

*everyone loves a metric!*

**dashboard**

*looks delightful on a flatscreen*

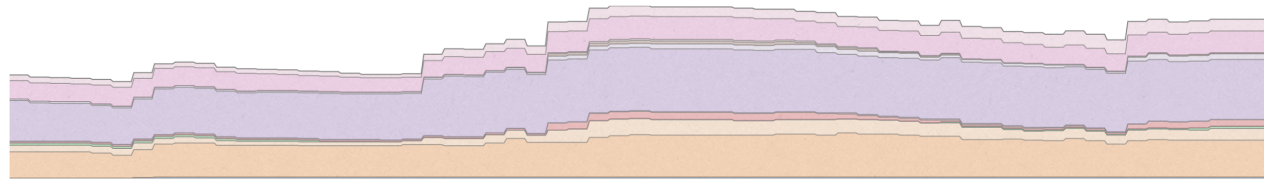
# what else does it do? : dashboard



vmpooler.delivery.puppetlabs.net

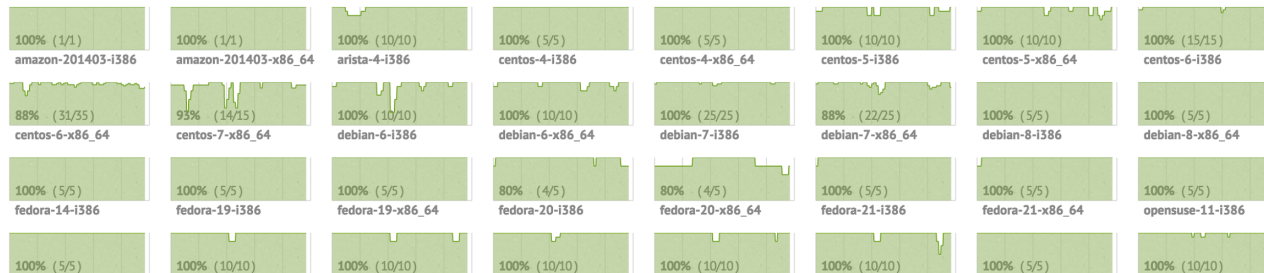
0 294 13 0 729 1036  
waiting to die running tests booting up being cloned ready and waiting total # of VMs

VMS RUNNING TESTS



amazon arista centos debian fedora opensuse oracle osx  
redhat scientific sles solaris ubuntu win

INDIVIDUAL POOL CAPACITY / FULLNESS STATUS



# what else does it do? : dashboard



vmpooler.delivery.puppetlabs.net

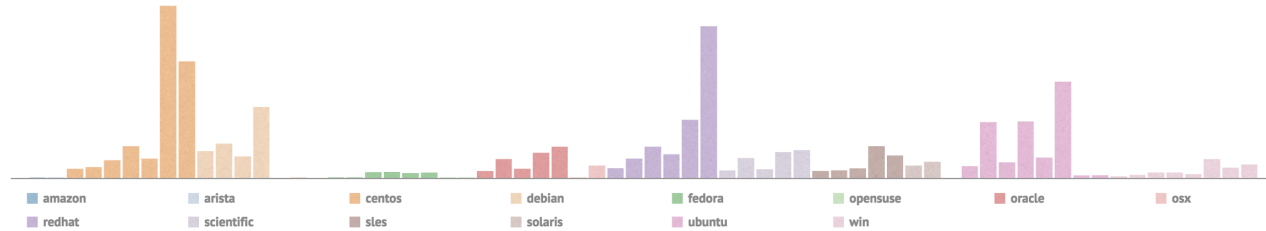
19,319  
cloned this week

mon.  
most clones

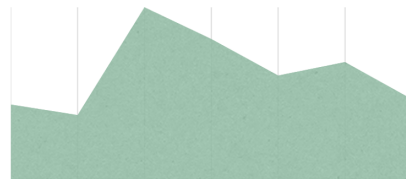
8.3s  
clone time average

119.3s  
boot time average

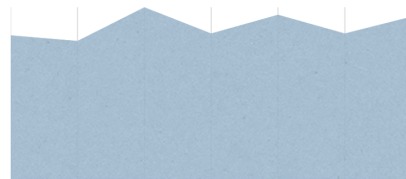
WEEKLY CLONE COUNTS BY POOL



WEEKLY TRENDS BY DAY



daily provision count



clone time ( average )



boot time ( average )

# what's next?

## tagging

*free-form*

*audit usage by user, build name, beaker version, etc.*

## authentication and API tokens

*configurable anonymous VS authenticated options*

*VM lifetime (TTL), etc.*

## other backends

*openstack? ec2?*

*something other than Redis?*

# questions?

 <https://github.com/puppetlabs/vmpooler>