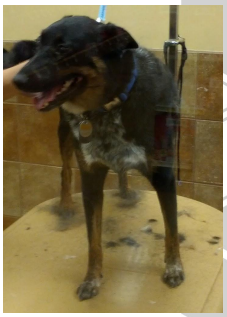
Today we are going to talk about:
What WebPageTest is and it's architecture
Some neat things about WebPageTest and what it can do
The challenges I faced with both WebPageTest and SaltStack.
But first, a little about me...

About Me

rackspace
the #1 managed cloud company

bitcoin

openstack



Linuturk on Twitter & Github onitato.com

Plug the Salt Master template.
Justin Phelps, SSCE
DevOps Engineer at Rackspace
Configuration management and CI/CD are my passions these days. My team maintains the Heat orchestration templates at Rackspace.
So what is WebPageTest?

Follow Along

States: github.com/linuturk/webpagetest

Slides: onitato.com

About WebPageTest

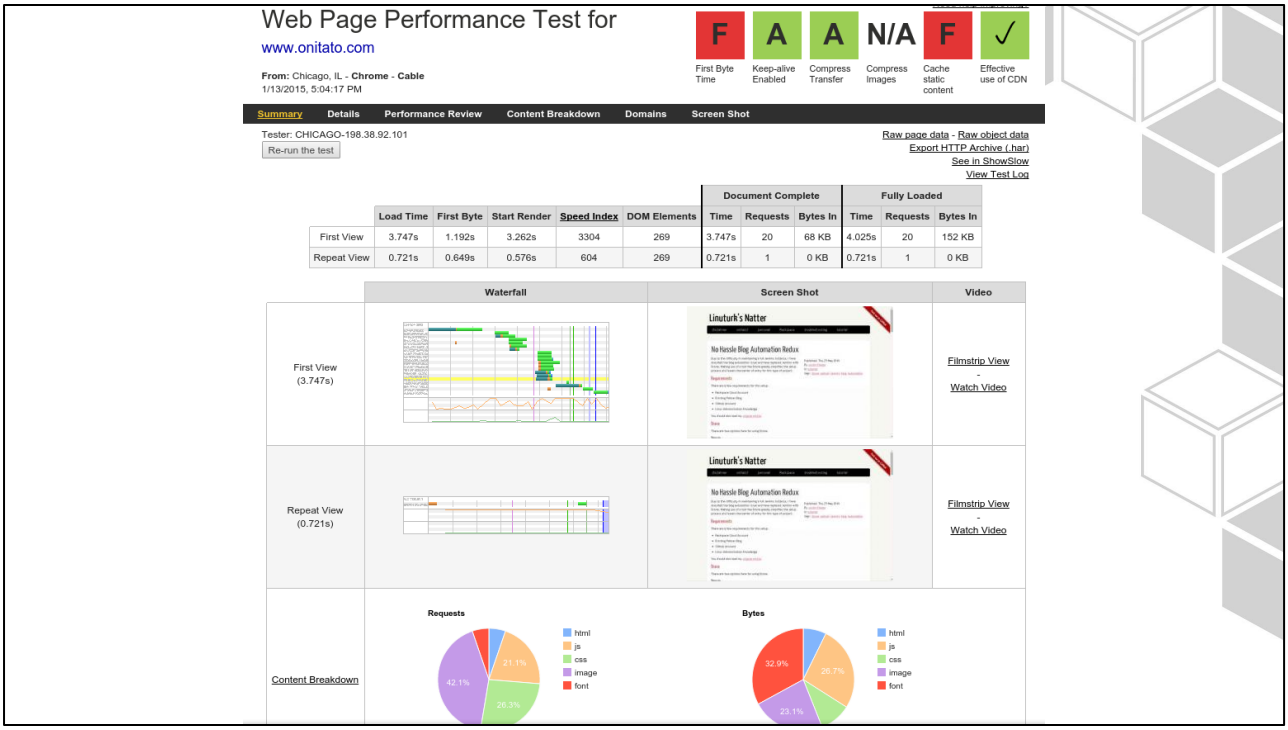
WEBPAGETEST

open source initiative

WebPageTest is an open source tool used to test the performance of websites on various browsers and devices.
Originally written at AOL and open sourced in 2008.
Source code is hosted in Github. Open Source!
The WPO Foundation runs the public instance.

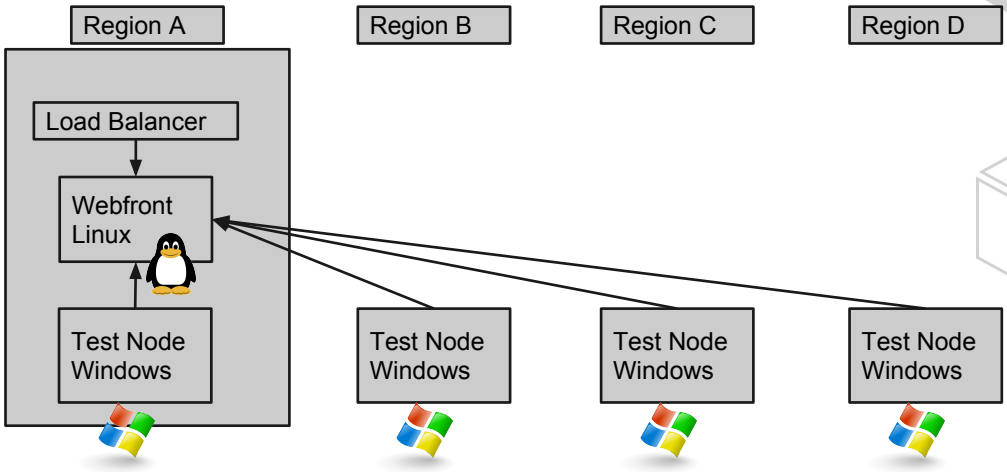
Browsers / Devices

The browsers shown here.
Android Phones and Tablets
let’s see an example test result.



Video of the site loading
Grading scale for major categories
Waterfall graph of the page load
Full API access to results
http://www.webpagetest.org/result/150113_E7_18YJ/

Private Instance Architecture



Architecture of the private instance we have deployed internally.
Design limits you to a single Linux front end node.
Windows nodes poll the PHP app on the front node for work.
Test history and results are stored on the front node. There's potential for a backup front node, but test history would have to be sync'd.

The Easy Parts

Front end was a fairly basic PHP application on Linux. Behind nginx. Basic stuff for SaltStack.
Once installed and configured correctly, the WebPageTest agents themselves handle automatic updating of themselves and the browsers they have configured.
Salt-cloud is next

Salt-Cloud

Providers

Profiles

Maps

Salt Cloud - Provider

```
my-rackspace-config:
  # Set the location of the salt-master
  #
  minion:
    master: <IP or Hostname>

  # Configure Rackspace using the OpenStack plugin
  #
  identity_url: 'https://identity.api.rackspacecloud.com/v2.0/tokens'
  compute_name: cloudServersOpenStack
  protocol: ipv4

  # Set the compute region:
  #
  compute_region: IAD

  # Configure Rackspace authentication credentials
  #
  user: <username>
  tenant: <tenant>
  apikey: <apikey>

  provider: openstack
```

Don’t forget to set the IP or DNS name for your master.
Rackspace Cloud in IAD.

Salt Cloud - Profile

```
wpt-linux:
    provider: my-rackspace-config
    size: 2 GB Performance
    image: Ubuntu 14.04 LTS (Trusty Tahr) (PVHVM)
wpt-windows:
    provider: my-rackspace-config
    size: 4 GB Performance
    image: 59901295-8ae6-4ab4-a697-01261799c8ac
    win_installer: /root/Salt-Minion-2014.1.7-AMD64-Setup.exe
    files:
      'C:\cloud-automation\bootstrap.cmd':
        /root/bootstrap.cmd
    grains:
      wpt_location: Location
      wpt_label: Label
      wpt_group: Group
      wpt_browsers: "Chrome,Firefox,IE"
```

Note the win_installer and files options.
Grains are specific to web page test. You'll want a separate profile for each region you run.
Let's look at that bootstrap.cmd file.

Salt Cloud - Bootstrap

```
netsh advfirewall firewall add rule name="SaltBootstrap" dir=in protocol=TCP localport=445 action=allow
```

```
$CurrentVal = Get-NetFirewallRule -DisplayName SaltBootstrap

if ($CurrentVal.Enabled -eq "True") {
  Disable-NetFirewallRule -DisplayName SaltBootstrap
  Write-Output "changed=yes comment='Port 445 Disabled.'"
} Else {
  Write-Output "changed=no comment='Port 445 Already Disabled.'"
}
```

Server Message Block (SMB)
Used server personality on the cloud to autorun this command when the server was created. You could also bake this into an image.
salt-cloud can't communicate with the server without this port.
Close it as part of your automation!

Salt Cloud - Map File

```
wpt-linux:
  - web01.webpagetest.example.com
  - web02.webpagetest.example.com
wpt-windows:
  - win01.webpagetest.example.com
  - win02.webpagetest.example.com
```

Map files are super useful for standing up infrastructure.
Let’s look at the Pillar data for this deployment next.

Pillar

```
webpagetest:
  sitename: "webpagetest.example.com"
  zipurl: "https://github.com/WPO-Foundation/webpagetest/releases/do
  zipsha: "sha256=19ee9df78205f99153c6fa80b6e6e98394cc680c4a9e13b31b
  http: "apache2" # nginx or apache2
  php: "php5" # php5 or php5-fpm
  win:
    user: "webpagetest"
    pass: "changeme"
    install_dir: 'C:\webpagetest'
    temp_dir: 'C:\wpttemp'
    zip_file: "webpagetest_2.15.zip"
  packages:
    - imagemagick
    - libimage-exiftool-perl
    - libjpeg-turbo-progs
    - php5-curl
    - php5-gd
    - php5-sqlite
    - php5-apcu
    - python-software-properties
    - unzip
```

example pillar data
sets values to be used later in various states, scripts, and files
Sitename is the dns name of the front end servers
zipurl zipsha ensure a good installer
can switch between nginx/apache php5/php5-fpm
win: includes settings for the windows portion, including the auto logon username and password.
packages: various dependencies for the webpagetest php front end.

Pillar

```
23  phpvars:
24    upload_max_filesize: "12M"
25    post_max_size: "12M"
26    memory_limit: "512M"
27    fpm: "unix:/var/run/php5-fpm.sock"
28  settings:
29    product: "WebPageTest"
30    contact: "contact@domain.com"
31    maxruns: 9
32    allowPrivate: 0
33    key: "changeme" #Do not use an & in your key
34  locations:
35    - name: TestLocation
36      label: TestLabel
37      group: TestGroup
38      browsers: "Chrome,Firefox,IE"
39      drivers:
40        - wptdriver
41        - urlblast
42      default: True
```

Bottom half of the example pillar data
tuning php vars for the front end php app
settings for webpagetest itself.
Locations (list of hashes that define all available locations for a test node)

Powershell States

```
close-port-445:
  cmd.script:
    - source: salt://webpagetest/powershell/Set-ClosePort445.ps1
    - shell: powershell
    - stateful: True
```

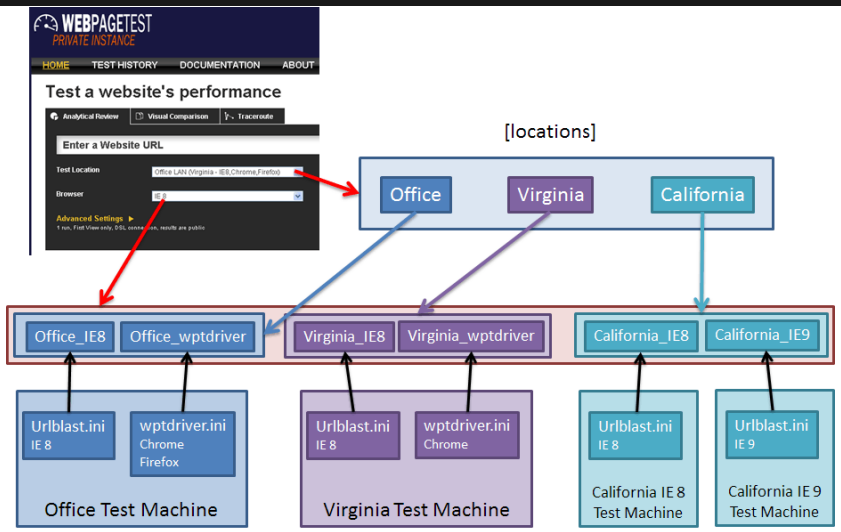
```
$CurrentVal = Get-NetFirewallRule -DisplayName SaltBootstrap

if ($CurrentVal.Enabled -eq "True") {
  Disable-NetFirewallRule -DisplayName SaltBootstrap
  Write-Output "changed=yes comment='Port 445 Disabled.'"
} Else {
  Write-Output "changed=no comment='Port 445 Already Disabled.'"
}
```

same as other cmd.scripts but set shell to ‘powershell’
take note of the use of stateful and the output written in the powershell - Best Practice!

WebPageTest Challenges

Configuring Locations
Windows Server + DummyNET Driver
Windows Automation
Mobile devices
The ampersand bug

Configuring Locations



This is from the WebPageTest documentation, and is there to explain how Locations should be configured.
Confused? So was I.
The basic idea is that all locations are defined on the Linux front end.
Then, you match your location settings on each Windows test node to the appropriate configuration on the front end.
I abstracted this into grains and pillar.

Configuring Locations

Pillar	Grains
webpagetest.locations.name	wpt_location
webpagetest.locations.label	wpt_label
webpagetest.locations.group	wpt_group
webpagetest.locations.browsers	wpt_browsers

```
locations:
  - name: TestLocation
    label: TestLabel
    group: TestGroup
    browsers: "Chrome,Firefox,IE"
    drivers:
      - wptdriver
      - urlblast
    default: True
```

```
wpt-windows:
    provider: my-rackspace-config
    size: 4 GB Performance
    image: 59901295-8ae6-4ab4-a697-01261799c
    win_installer: /root/Salt-Minion-2014.1.
    files:
      'C:\cloud-automation\bootstrap.cmd':
        /root/bootstrap.cmd
    grains:
      wpt_location: Location
      wpt_label: Label
      wpt_group: Group
      wpt_browsers: "Chrome,Firefox,IE"
```

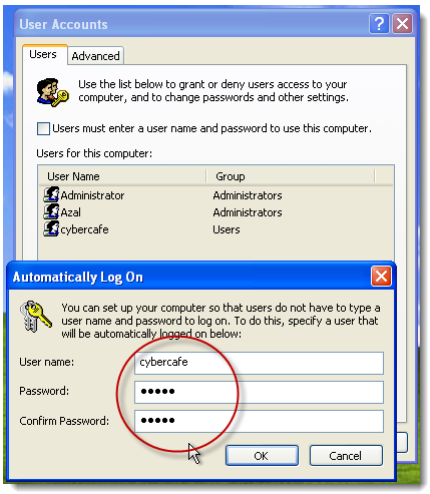
Define locations in Pillar and iterate over this list to write the front end location configuration.
Configure grains in salt-cloud and write their configs using grain data.
As long as those entries in Pillar match the same Grain data, your locations should work.

Xen + DummyNET Driver

UPPSALA
UNIVERSITET

Can’t perform traffic shaping without this DummyNET Driver.
The Xen hypervisor isn’t compatible without special changes that aren’t allowed on the Rackspace Public Cloud. (AWS as well.)
mindinst.exe allowed me to automate the installation of the DummyNET network binding. There isn’t a way to do this with pure PowerShell.
Uppsala University is a research university in Uppsala, Sweden, and is the oldest university in Sweden, founded in 1477.

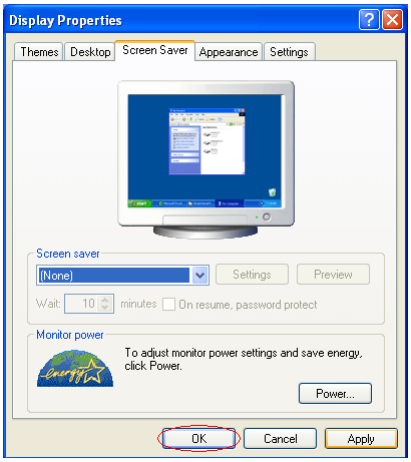
Auto Logon




```powershell
$LogonPath = 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon'
$LastUser = 'HKLM:\Software\Microsoft\Windows\CurrentVersion\Authentication\LogonUI'
$Domain = "{{ grains.host }}"
$Username = "{{ pillar['webpagetest']['win']['user'] }}"
$Password = "{{ pillar['webpagetest']['win']['pass'] }}"

$CurrentVal = Get-ItemProperty -Path $LogonPath -Name AutoAdminLogon

If ($CurrentVal.AutoAdminLogon -eq "1") {
  $CurrentUser = Get-ItemProperty -Path $LogonPath -Name DefaultUserName
  $CurrentPass = Get-ItemProperty -Path $LogonPath -Name DefaultPassword

  If ($CurrentUser.DefaultUserName -ne $Username -Or $CurrentPass.DefaultPassword -ne $Password) {
    Set-ItemProperty -Path $LogonPath -Name DefaultUserName -Value "$Username"
    Set-ItemProperty -Path $LogonPath -Name DefaultPassword -Value "$Password"
    Write-Output "changed=yes comment='Credentials Updated.'"
  } Else {
    Write-Output "changed=no comment='AutoLogon already enabled.'"
  }
} Else {
  Set-ItemProperty -Path $LogonPath -Name AutoAdminLogon -Value "1"
  Set-ItemProperty -Path $LogonPath -Name DefaultDomainName -Value "$Domain"
  New-ItemProperty -Path $LogonPath -Name DefaultUserName -Value "$Username"
  New-ItemProperty -Path $LogonPath -Name DefaultPassword -Value "$Password"
  New-ItemProperty -Path $LogonPath -Name DontDisplayLastUserName -Value 1
  Set-ItemProperty -Path $LastUser -Name LastLoggedOnUser -Value "$Username"
  Set-ItemProperty -Path $LogonPath -Name LastUsedUsername -Value "$Username"
  Write-Output "changed=yes comment='AutoLogon enabled.'"
}
```

Auto logon and screensavers is for the video capture that happens while a page is being loaded.

Disable Screensavers




```powershell
$Path = 'HKCU:\Control Panel\Desktop'

Try {
  $CurrentVal = Get-ItemProperty -Path $Path -Name ScreenSaveActive
  Write-Output $CurrentVal
} Catch {
  $CurrentVal = False
} Finally {
  if ($CurrentVal.ScreenSaveActive -ne 0) {
    Set-ItemProperty -Path $Path -Name ScreenSaveActive -Value 0
    Write-Output "changed=yes comment='Screensaver Disabled.'"
  } Else {
    Write-Output "changed=no comment='Screensaver Already Disabled.'"
  }
}
```

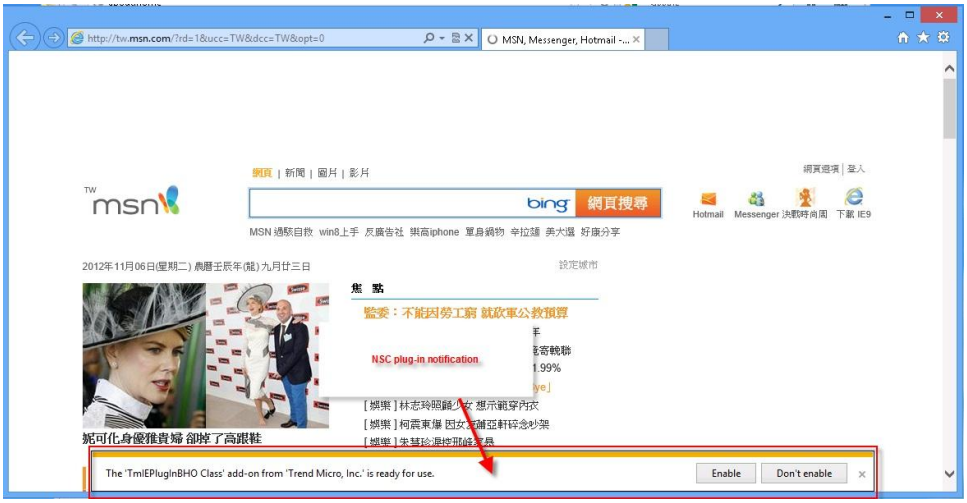
Auto logon and screensavers is for the video capture that happens while a page is being loaded.

Disable Shutdown Tracker


```powershell
$Path = 'HKLM:\Software\Microsoft\Windows\CurrentVersion\Reliability'

Try {
  $CurrentVal = Get-ItemProperty -Path $Path -Name ShutdownReasonUI
  Write-Output $CurrentVal
} Catch {
  $CurrentVal = False
} Finally {
  if ($CurrentVal.ShutdownReasonUI -ne 0) {
    New-ItemProperty -Path $Path -Name ShutdownReasonUI -Value 0
    Write-Output "changed=yes comment='Shutdown Tracker Disabled.'"
  } Else {
    Write-Output "changed=no comment='Shutdown Tracker Already Disabled.'"
  }
}
```

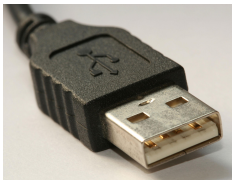
Auto reboots are performed by the webpagetest agent when necessary for updates.

IE Browser Plugin



There isn't a way through the registry or PowerShell to authorize a browser plugin. You can only install them. They won't actually work until they are authorized when the browser first launches.

Mobile Devices



Android devices require USB debugging and a physical connection via USB to the testing node.
Our deployment was done on the Rackspace Cloud, so this wasn't possible.

The Ampersand (&) Bug

&

This character gave me hell for about an entire day. Care to guess how?

The Ampersand (&) Bug

```
/work/getwork.php?
shards=1&location=Virginia_wptdriver&key=secret1
23&software=wpt&version=2.15.0.183
&ver=183&pc=COMPUTERNAME&dns=69.20.0.164-
69.20.0.196&freedisk=17.036&GPU=0
```

What if I show you this line from the logs ...

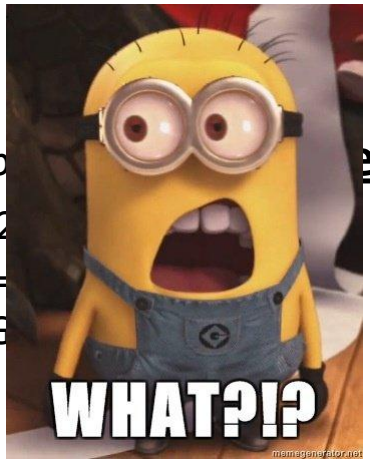
The Ampersand (&) Bug

/work/getwork.php?
shards=1&location=Virginia_wptdriver&**key=secret123**&software=wpt&version=2.15.0.183
&ver=183&pc=COMPUTERNAME&dns=69.20.0.164-
69.20.0.196&freedisk=17.036&GPU=0

WAT?

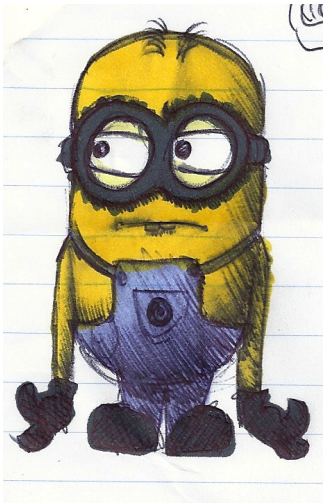
The Ampersand (&) Bug

```
/work/getwork.php?
shards=1&location=Virginia_wp                    ecr
et123&software=wpt&version=2
&ver=183&pc=COMPUTERNAME&dns=
69.20.0.196&freedisk=17.036&G
```



I generate random secure passwords for my work. The one I generated to act as the secret key had an ampersand. Since work requests are passed using a PHP GET request, everything is passed in the URL of the request. The ampersand effectively cut my key in half, making the request fail.

SaltStack Challenges



salt-cloud windows bootstrapping
available windows modules
learning powershell

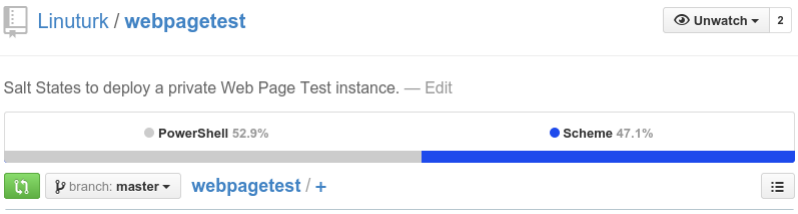
Windows Salt-Cloud

Bootstrapping Windows with salt-cloud today is a weird experience:

- ~~Use Password from OpenStack API - PR14748~~
- Fingerprint Mismatch, Map File Broke - #14593
- Discussion on new bootstrap method - #15984

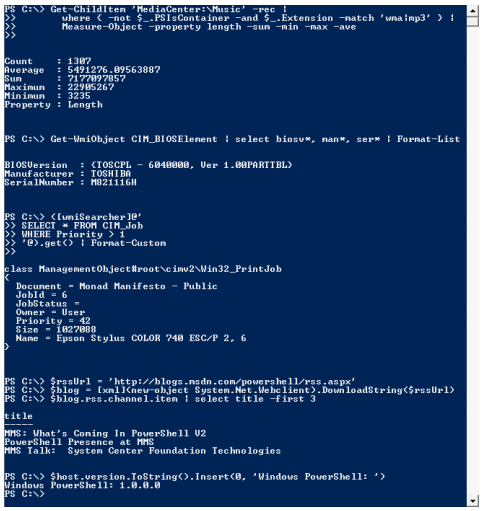
The PR was authored by me and fixed an immediate problem with passwords.
Recently backported to 2014.7 and has been merged! YAY!
You still can't use a map file to provision multiple windows servers at the same time.
The Salt Master fingerprint gets corrupted on the minion, and it never connects properly.
There's a larger discussion going on about replacing winexe with another Windows remote execution tool.

Windows State Modules



Over half the code written for this project was PowerShell code versus the YAML (mis-identified as Scheme by Github).
A lot of this code could be replaced with a Windows Registry State Module. Hack Day project?

Learning PowerShell



Actually not that bad. Was a positive experience overall.
Lucky that I sit right next to 3 PowerShell developers at my office.

Try WebPageTest

Give WebPageTest a try using:

the public instance at webpagetest.org
Salt to spin up your own private architecture
Rackspace Orchestration Template (Heat)

We also have a Salt Master template!

Questions?

Submit Feedback!

Slides at: onitato.com

States and PowerShell at:
www.github.com/linuturk/webpagetest