

Automated Software and Configuration Deployment Using **PowerShell** and **Chocolatey**

“just create a package for it”

Manfred Wallner

- software developer (in automotive industry)
- hobby photographer
- travel enthusiast

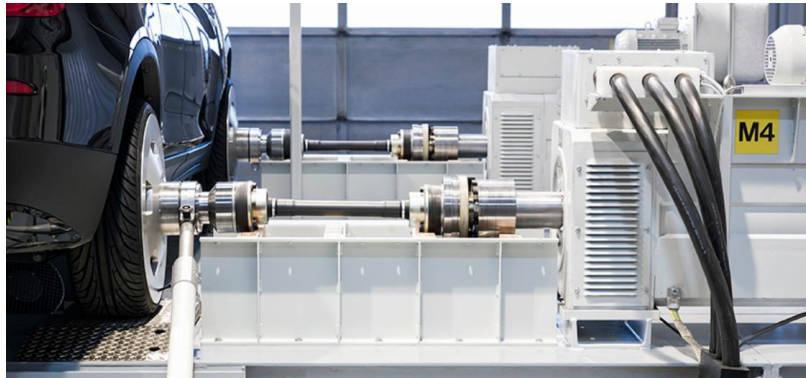
Chocolatey user since summer 2016

volunteer Chocolatey package moderator



KS Engineers

- 40+ years in business
- 450+ employees
- testbenches for automotive industry
 - automotive testing / industrial automation
 - hard + software



THE STORY BEGINS

... about 3 years ago, in a country not that far away ...

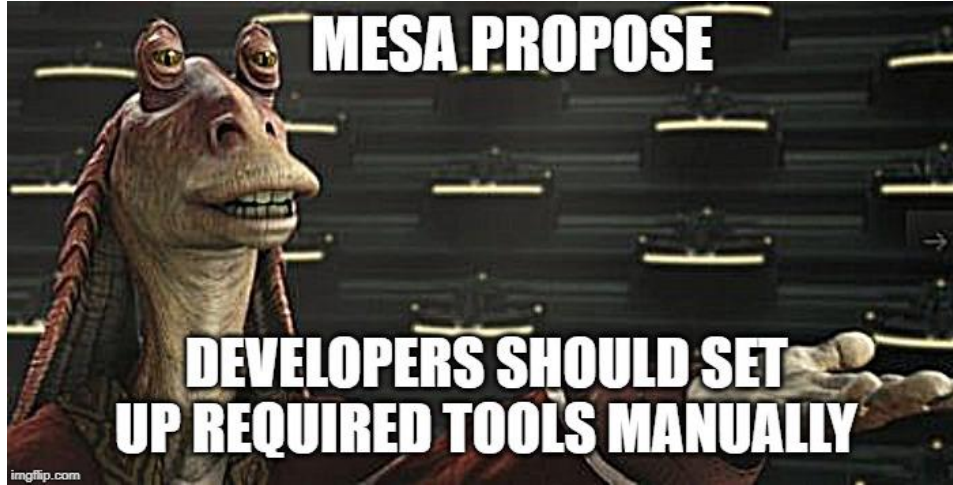
we have been upgrading our dev boxes

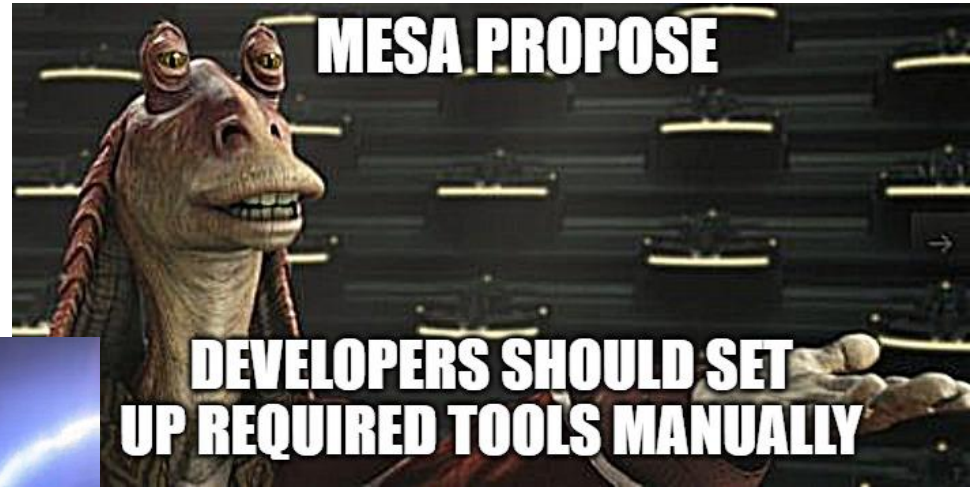
fancy i7s, TONs of RAM, fast SSDs ...

we have been upgrading our dev boxes

... the roll-out „collided“ with a couple of tool & configuration upgrades ...

Automated Software and Configuration Deployment?





dev boxes,
servers should
„align“



drifting state vs. desired state

- software
 - version
 - configuration
- infrastructure
 - # hosts / nodes (dynamic scaling?)
 - configuration



drifting state?

- “what did we do?”
- “it should not do that.”
- “try re-running the admin wizard.”
- “did it work yesterday?”
- “try it on another machine.”



drifting state vs. desired state

- software
 - version
 - configuration
- infrastructure
 - # hosts / nodes (dynamic scaling?)
 - configuration



desired state?

- everything is in working condition
- full knowledge of the config
- transparent, known state
- have a complete audit log



REACHING A GOAL

REQUIRES PROPER TOOLS

golden images ...



the good and the bad of golden images

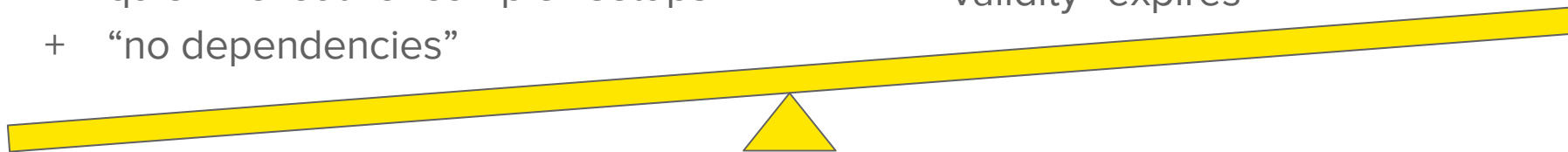
- + easy to understand
- + quick “rollout” of complex setups
- + “no dependencies”
- hard to “change” (updates)
- versioning?
- “validity” expires



do the pro's weigh out the con's?

- + easy to understand
- + quick “rollout” of complex setups
- + “no dependencies”

- hard to “change” (updates)
- versioning?
- “validity” expires



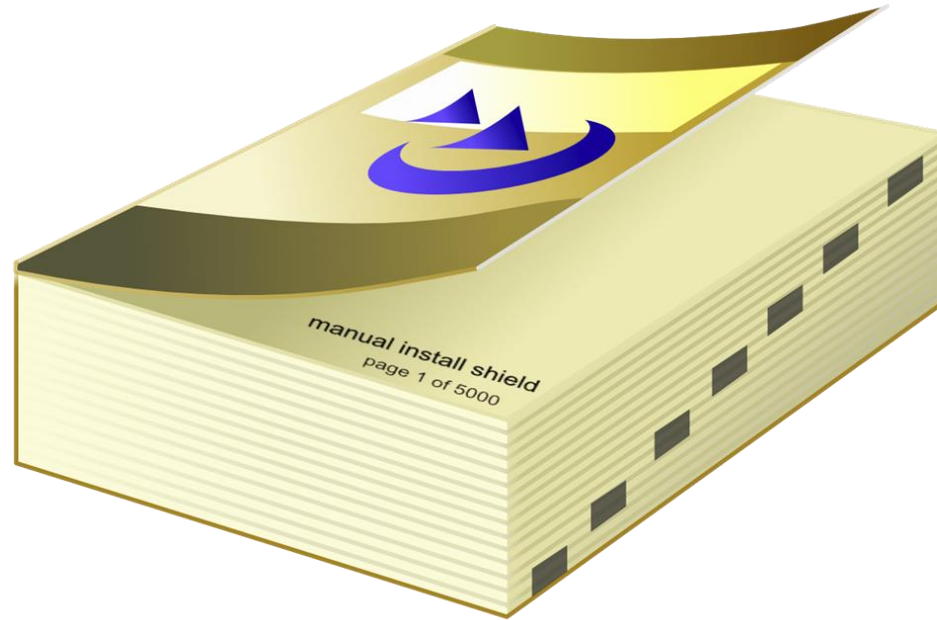
do the pro's weigh out the con's?

- + easy to understand
- + quick “rollout” of complex setups
- + “no dependencies”

- hard to “change” (updates)
- versioning?
- “validity” expires

... nope ...

another extreme ...



let's automate it



automated software mgmt/deployment

- SCCM
- Symantec Client Management Suite
- ZENworks
- *Chef / Puppet / Ansible*
- Oinstall
- Ninite
- Chocolatey
- PowerShell + DSC ... ?





most tools have ONE thing in common

logic is expressed in PowerShell

PowerShell?



Can ...

- do “everything” on Windows Desktop OS
- do **everything** on Windows Server / Azure

Provides ...

- powerful object-based I/O mechanisms
- rapid prototyping
- high extensibility and OO as well as functional paradigms

Feels ...

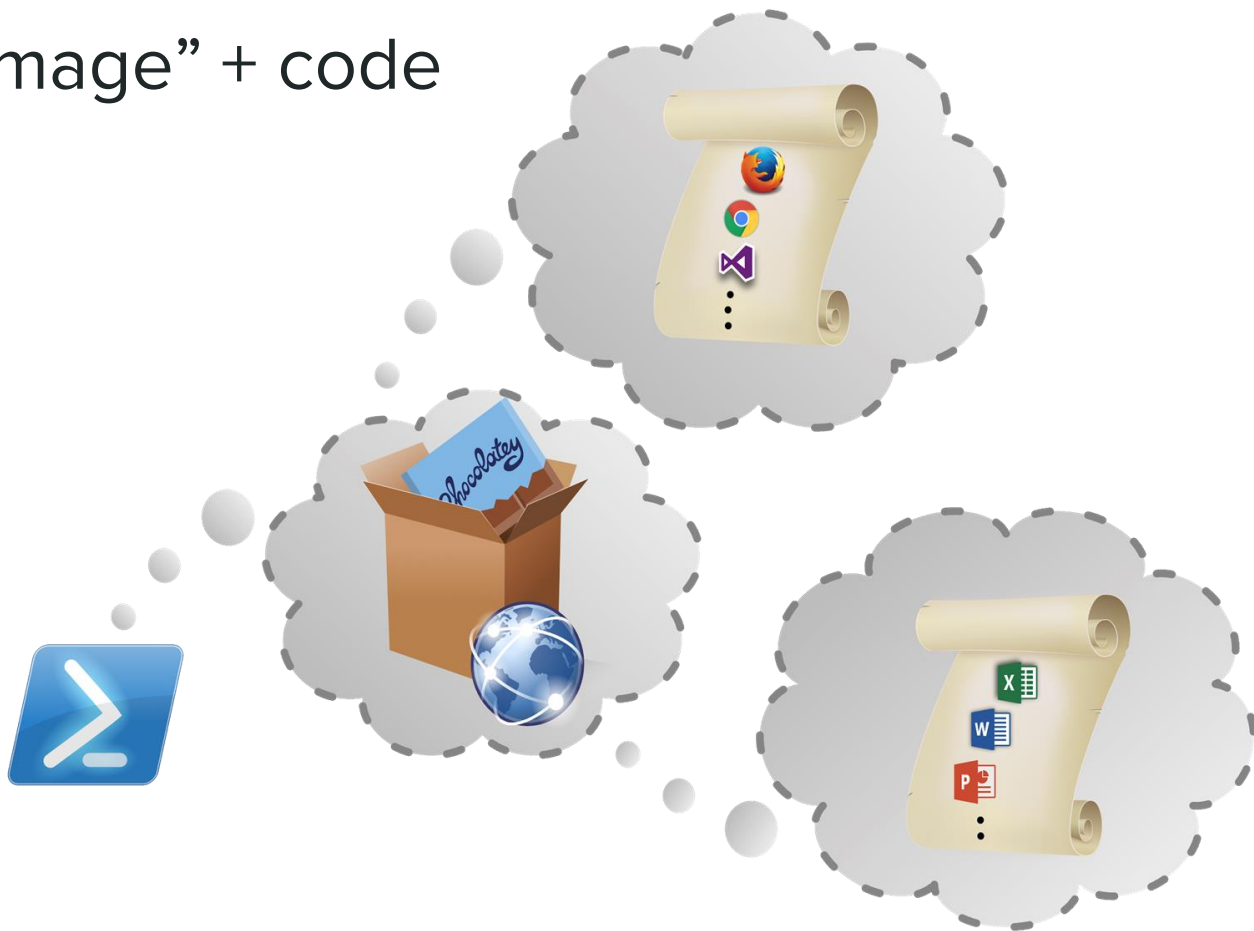
- **very different** from known tools / shells on first contact

PowerShell!

- de-facto standard for Windows System Automation
- lots of public domain know-how & how-to's
- also applicable for cloud & Linux environments



base “image” + code



Chocolatey?

- a real package manager, but on Windows!
- already large and growing community
- moderated community repository
- responsive “24/7” support
- package manager (base) = FOSS
- business friendly options available



what is a *Chocolatey* package?

- nupkg (zip + meta info)
- PowerShell
 - ChocolateyInstall.ps1
 - ChocolateyUninstall.ps1
 - ChocolateyBeforeModify.ps1



MONSTER SETUPS

require multiple reboot
take many dependencies



MONSTER SETUPS

require multiple reboot
take many dependencies

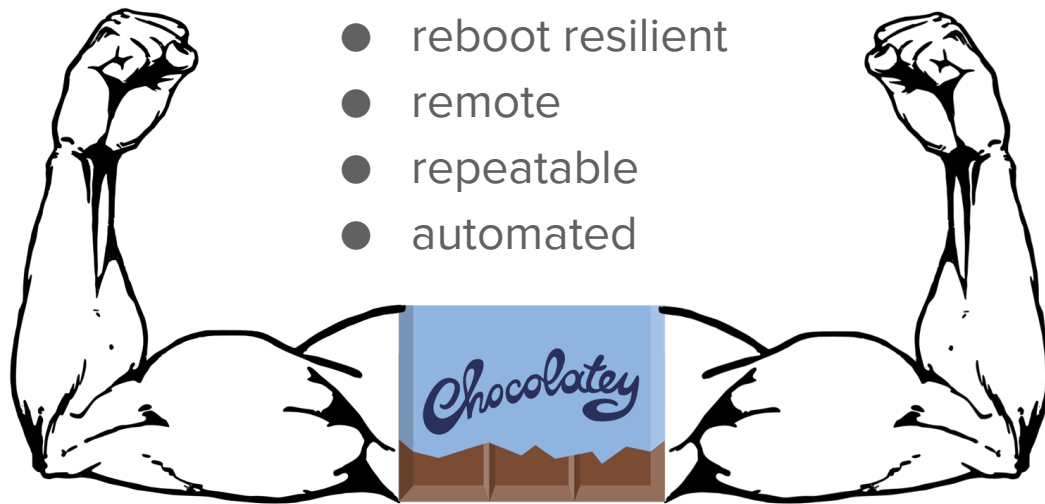


MONSTER SETUPS

require multiple reboot
take many dependencies



Boxstarter



- reboot resilient
- remote
- repeatable
- automated

<https://boxstarter.org/>

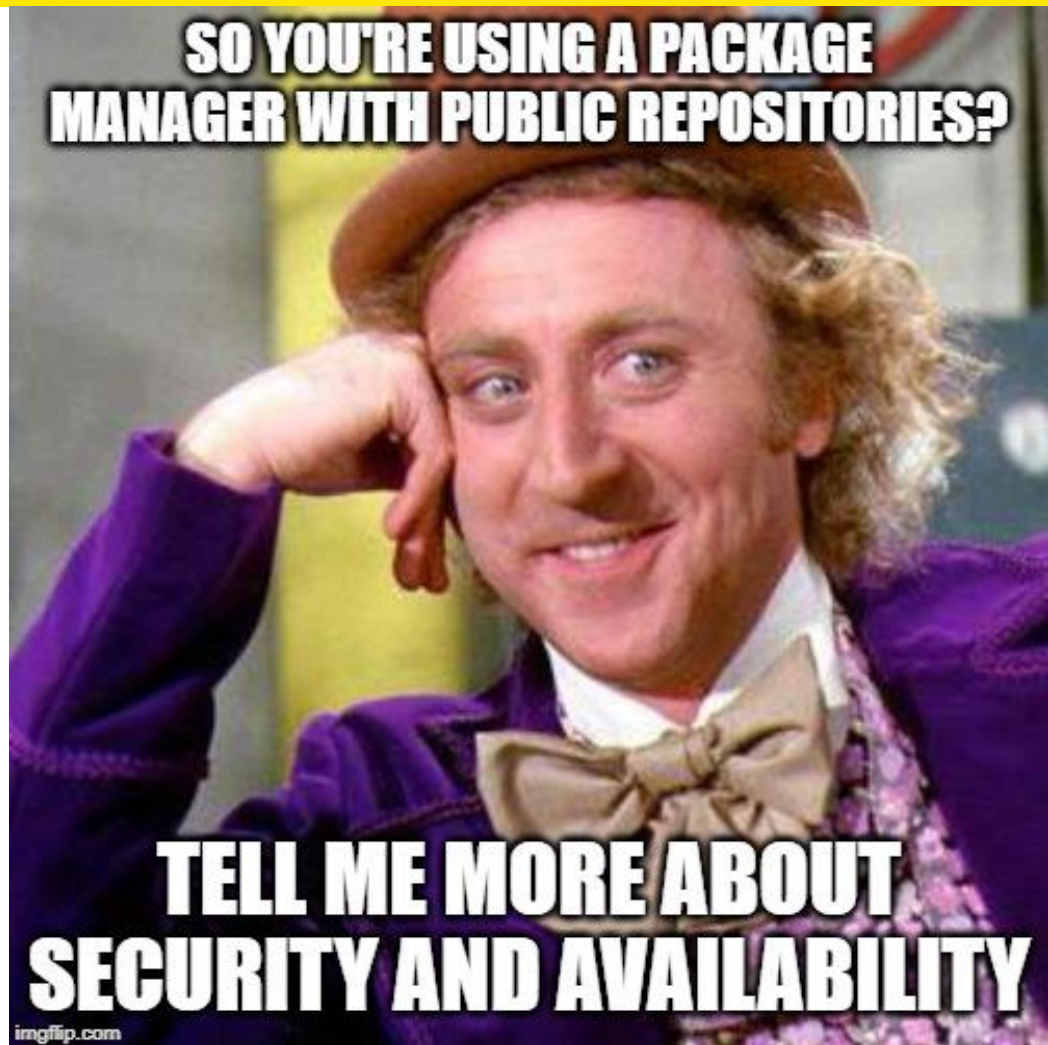
Boxstarter - remote install

```
PS> Install-BoxstarterPackage -ComputerName
starkillermain01 -PackageName blowup -Credential
$secretcreds
Boxstarter Version 2.11.0
(c) 2018 Chocolatey Software, Inc, 2012 - 2018 Matt Wrock. https://boxstarter.org

Boxstarter: Configuring local PowerShell Remoting settings...
Boxstarter: Configuring CredSSP settings...
Boxstarter: Testing remoting access on starkillermain01.....

. . .
```


security advisors
be like...



internal repository

- ProGet
- Artifactory
- Samba share
- ...



JFrog Artifactory



relying on the cloud?

- implicit & explicit dependencies
- the horror story of npm / maven / nuget / ...?
 - „left-pad“ (20 loc) – March 2016
- can your build / deployments run without...
 - internet?
 - docker images?



you don't wanna be =>



automated package internalization

```
PS> choco download --internalize $pkg `
      --resources-location="$uncshare\$pkg" `
      --source="$basefeed" --no-progress
```

...

automated package internalization

...

```
PS> $genpkg = (gci *.nupkg -recurse).FullName
```


...

automated package internalization

...


```
PS> $genpkg | % { choco push $_.FullName `
    --source="$targetserver" `
    --api-key="$apikey" -Verbose }
```


automated package internalization (with Jenkins #1)


 **Jenkins**


3


Jenkins > Chocolatey > Internalize_Pkgs > #8


 [Back to Project](#)


 [Status](#)


 [Changes](#)


 **Console Output**

 [View as plain text](#)

 [Edit Build Information](#)

 [Parameters](#)

 [Previous Build](#)


 **Console Output**

Progress:

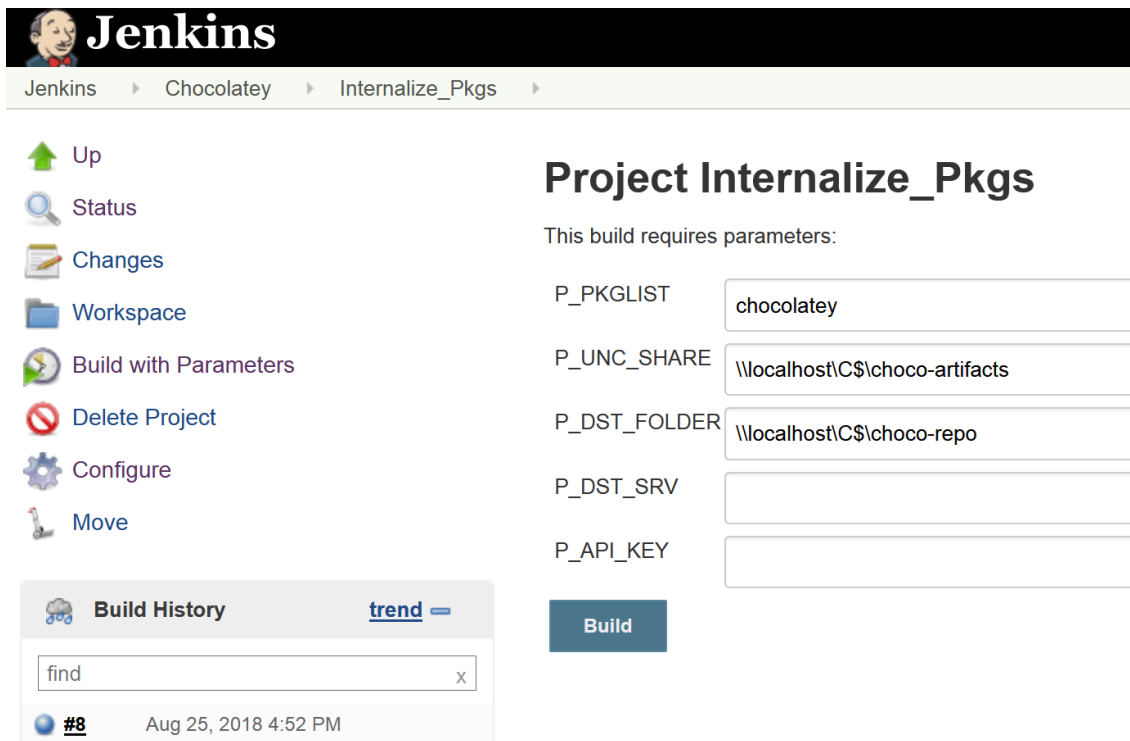
Started by user [anonymous](#)
Building in workspace C:\Program Files (x86)\Jenkins\workspace\Chocolatey\Internalize_Pkgs
[Internalize_Pkgs] \$ powershell.exe -NonInteractive -ExecutionPolicy Bypass "& 'C:\Windows\TEMP\jenkins736050888825954969.ps1'"
VERBOSE: Performing the operation "Create Directory" on target "Destination: C:\Windows\TEMP\chocointernalize".

Directory: C:\Windows\TEMP

Mode	LastWriteTime	Length	Name
d----	8/25/2018 4:52 PM		chocointernalize

Chocolatey v0.10.11 Business
Downloading existing package(s) to C:\Windows\TEMP\chocointernalize\download


automated package internalization (with Jenkins #2)



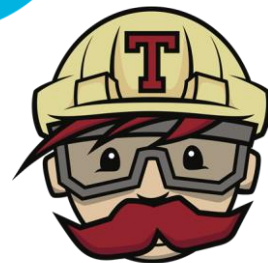
The screenshot displays the Jenkins web interface. At the top, the Jenkins logo and name are visible. Below the navigation bar, the breadcrumb trail shows 'Jenkins > Chocolatey > Internalize_Pkgs'. On the left sidebar, there are links for 'Up', 'Status', 'Changes', 'Workspace', 'Build with Parameters', 'Delete Project', 'Configure', and 'Move'. The main content area is titled 'Project Internalize_Pkgs' and states 'This build requires parameters:'. Below this, there are input fields for 'P_PKGLIST' (containing 'chocolatey'), 'P_UNC_SHARE' (containing '\\localhost\C\$\choco-artifacts'), 'P_DST_FOLDER' (containing '\\localhost\C\$\choco-repo'), 'P_DST_SRV' (empty), and 'P_API_KEY' (empty). A 'Build' button is located below these fields. At the bottom left, there is a 'Build History' section with a search bar containing 'find' and a list of builds, with the most recent being '#8' dated 'Aug 25, 2018 4:52 PM'.

Package Name	Internalize / Manual	Internalize / Jenkins
VirtualBox	~ 15 minutes	~ 1 minute
Firefox	~ 5 minutes	~ 30 seconds
Wireshark	~ 10 minutes	~ 30 seconds
WinPcap	~ 5 minutes	~ 20 seconds



CI joining the party

- Powershell / “choco” CLI - integrates *everywhere*
 - AppVeyor
 - Travis
 - Win Task Scheduler
 - **Jenkins**
 - Plugin [build-with-parameters](#)
 - Plugin [PowerShell](#)
 - ...



Travis CI

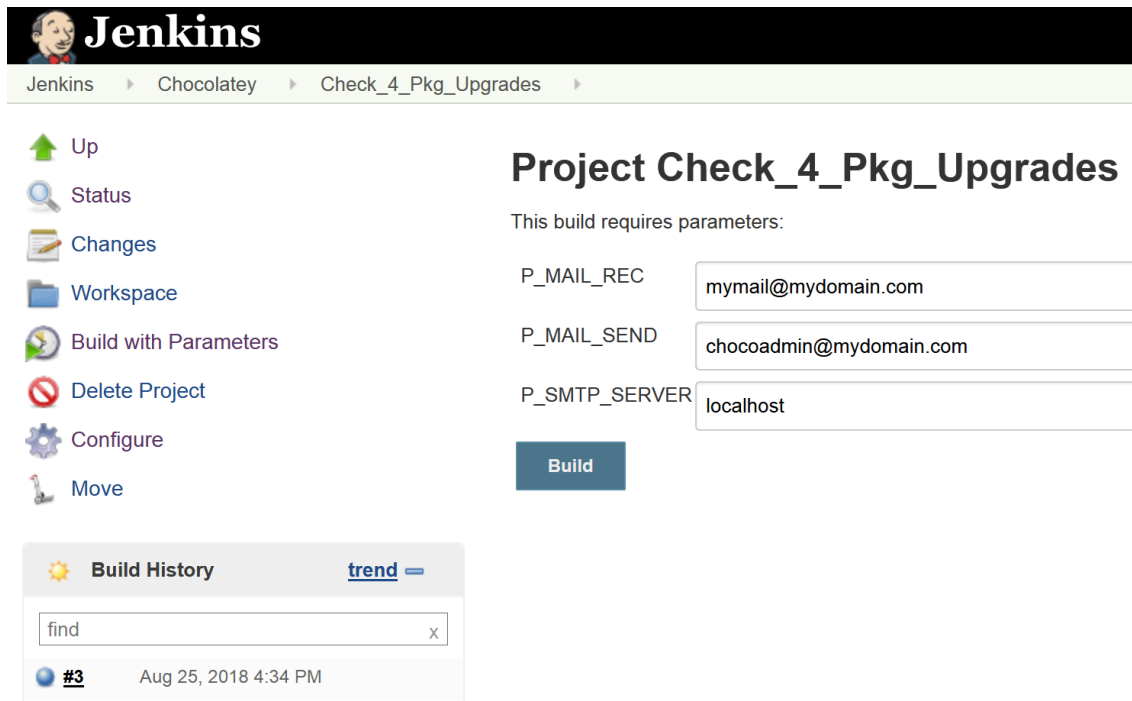


Jenkins

getting notified of possible updates (manual)

```
PS> choco outdated -r
AzurePowerShell|6.4.0|6.6.0|false
jdk10|10.0.1|10.0.2|false
notepadplusplus|7.5.7|7.5.8|false
...
```

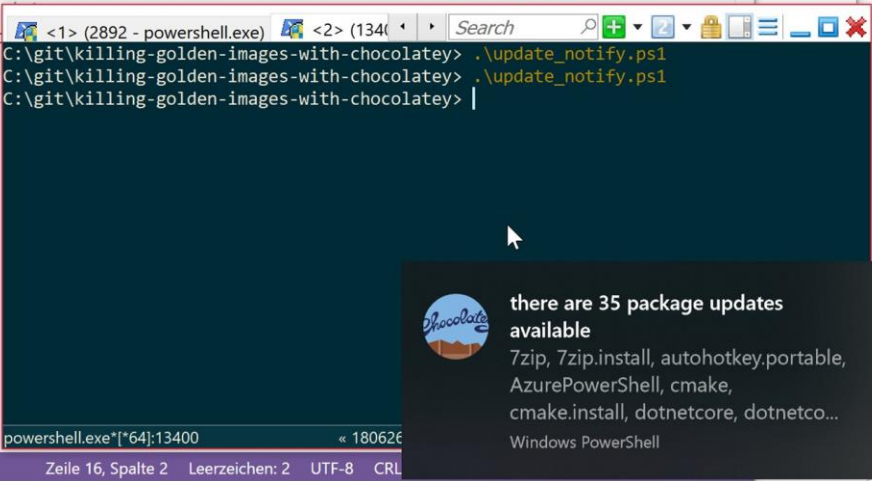
getting notified of possible updates (with Jenkins CI)



The screenshot shows the Jenkins web interface. At the top is the Jenkins logo and a breadcrumb trail: Jenkins > Chocolatey > Check_4_Pkg_Upgrades. On the left is a sidebar with icons and links for Up, Status, Changes, Workspace, Build with Parameters, Delete Project, Configure, and Move. The main area is titled 'Project Check_4_Pkg_Upgrades' and contains the text 'This build requires parameters:'. Below this are three input fields: 'P_MAIL_REC' with the value 'mymail@mydomain.com', 'P_MAIL_SEND' with the value 'chocoadmin@mydomain.com', and 'P_SMTP_SERVER' with the value 'localhost'. A blue 'Build' button is positioned below these fields. At the bottom left is a 'Build History' section with a search bar containing 'find' and a list of builds, with the first entry being '#3' dated 'Aug 25, 2018 4:34 PM'.

getting notified of possible updates (Win tasks)

```
$pkgText += "..."  
}  
New-BurntToastNotification -Text "there are $total package updates  
available",  
C:\git\kill
```



The screenshot shows a PowerShell terminal window with the following commands and output:

```
C:\git\killing-golden-images-with-chocolatey> .\update_notify.ps1  
C:\git\killing-golden-images-with-chocolatey> .\update_notify.ps1  
C:\git\killing-golden-images-with-chocolatey> |
```

A Windows toast notification is displayed over the terminal window. The notification has a blue circular icon with the word "chocolatey" and the text:

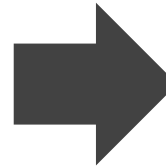
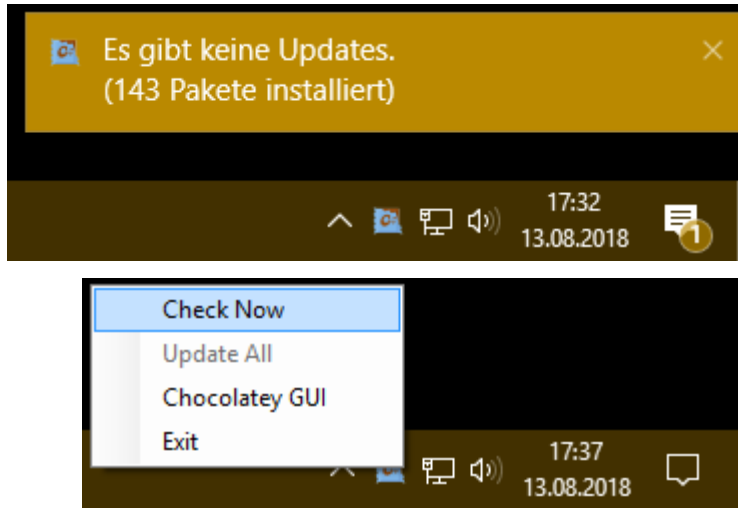
there are 35 package updates available
7zip, 7zip.install, autohotkey.portable,
AzurePowerShell, cmake,
cmake.install, dotnetcore, dotnetco...
Windows PowerShell



getting notified of possible updates

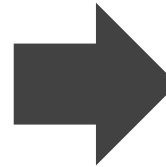
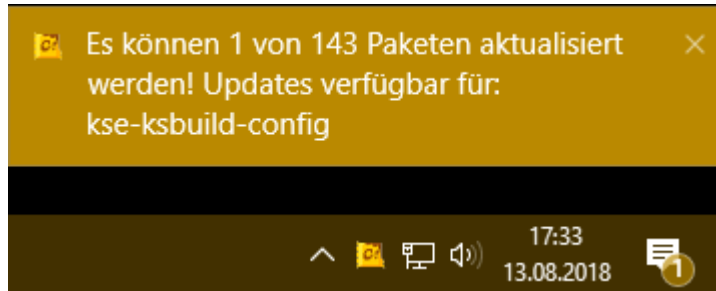
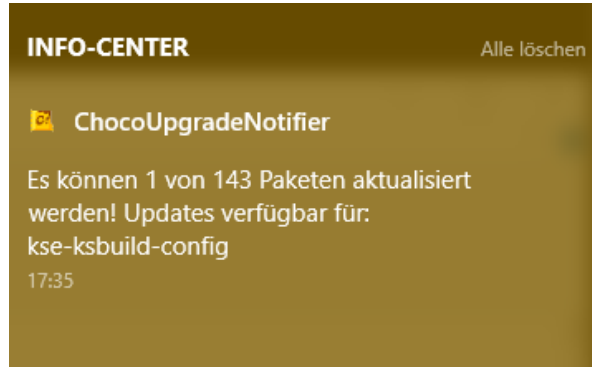


```
PS> choco install -y ChocoUpgradeNotifier
```



no updates available

getting notified of possible updates



1 update available

monitoring / alerting / statistics ?

- DIY – PowerShell logfile analysis
- DIY – PoSh => ElasticSearch + Kibana
- C4B – Chocolatey Central Management
- ... ?

monitoring / alerting / statistics ?

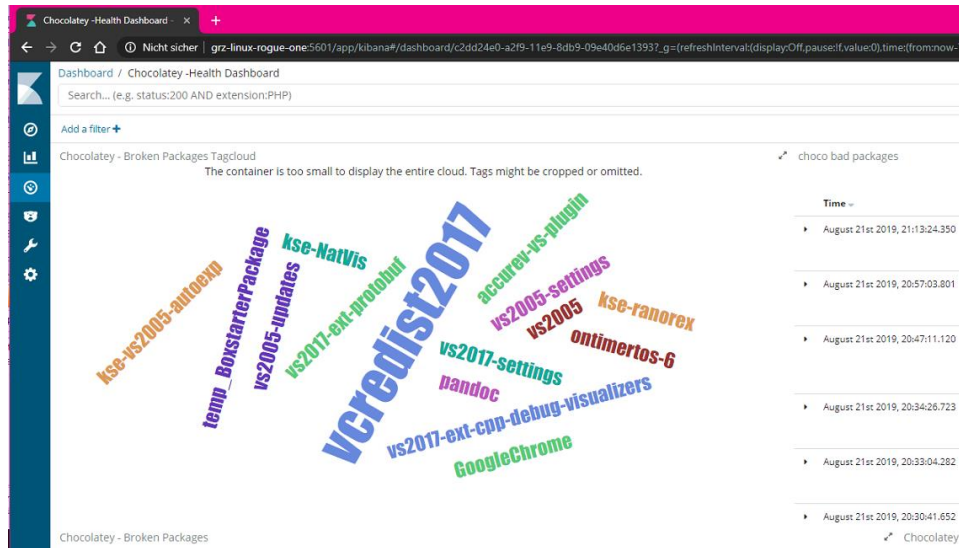
- DIY – PowerShell logfile analysis
- DIY – PoSh => ElasticSearch + Kibana
- C4B – Chocolatey Central Management
- ... ?

TESTPC12-PC	KS.SYSTEM	chocoupgradenotifier 3.3.0.0	21.08.2019 20:28:23	2019-08-21T20:28:23.8688622+02:00
TESTPC14	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	21.08.2019 13:39:54	2019-08-21T13:39:53.7267759+02:00
TESTPC18	KS.SYSTEM	chocoupgradenotifier 3.3.1.0	21.08.2019 20:28:18	2019-08-21T20:28:20.7334126+02:00
TESTPC19	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	21.08.2019 20:50:07	2019-08-21T20:50:07.6183334+02:00
TESTPC20	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	21.08.2019 21:14:02	2019-08-21T21:14:02.9346612+02:00
TESTPC21	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	21.08.2019 13:25:39	2019-08-21T13:25:39.76729+02:00
TORR04	KS.SYSTEM	chocoupgradenotifier 3.3.0.0	21.08.2019 20:47:50	2019-08-21T20:47:50.5262+02:00
tutica	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	21.08.2019 16:16:57	2019-08-21T16:16:57.7502189+02:00
uenluei3	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	09.08.2019 15:22:55	2019-08-09T15:22:55.0454702+02:00
volkmera	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	16.08.2019 11:56:19	2019-08-16T11:56:18.9464506+02:00
VOLLMERA2	KS.SYSTEM	chocoupgradenotifier 3.3.0.0	16.08.2019 13:59:44	2019-08-16T13:59:39.178486+02:00
VORAUERM2	KS.SYSTEM	chocoupgradenotifier 4.1.1.0	06.08.2019 16:41:04	2019-08-06T16:41:04.3461923+02:00
wallnerm4	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	14.08.2019 15:10:06	2019-08-14T15:10:06.3739388+02:00
wallnerm5	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	21.08.2019 20:37:53	2019-08-21T20:37:54.1066529+02:00
WIPFLERM2	KS.SYSTEM	chocoupgradenotifier 3.3.1.0	20.08.2019 06:04:49	2019-08-20T06:04:49.7112774+02:00
wipflerm2	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	21.08.2019 15:02:31	2019-08-21T15:02:27.7504958+02:00
woengoetterc3	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	21.08.2019 21:16:49	2019-08-21T21:16:49.4013179+02:00
woengoetterc4	KS.SYSTEM	chocoupgradenotifier 4.2.0.0	20.08.2019 15:53:55	2019-08-20T15:53:56.2136471+02:00
ZINTERLM	KS.SYSTEM	chocoupgradenotifier 4.1.1.0	21.08.2019 16:36:19	2019-08-21T16:36:19.8857688+02:00

chocoupgradenotifier is installed on 142 hosts

monitoring / alerting / statistics ?

- DIY – PowerShell logfile analysis
- DIY – PoSh => ElasticSearch + Kibana
- C4B – Chocolatey Central Management
- ... ?



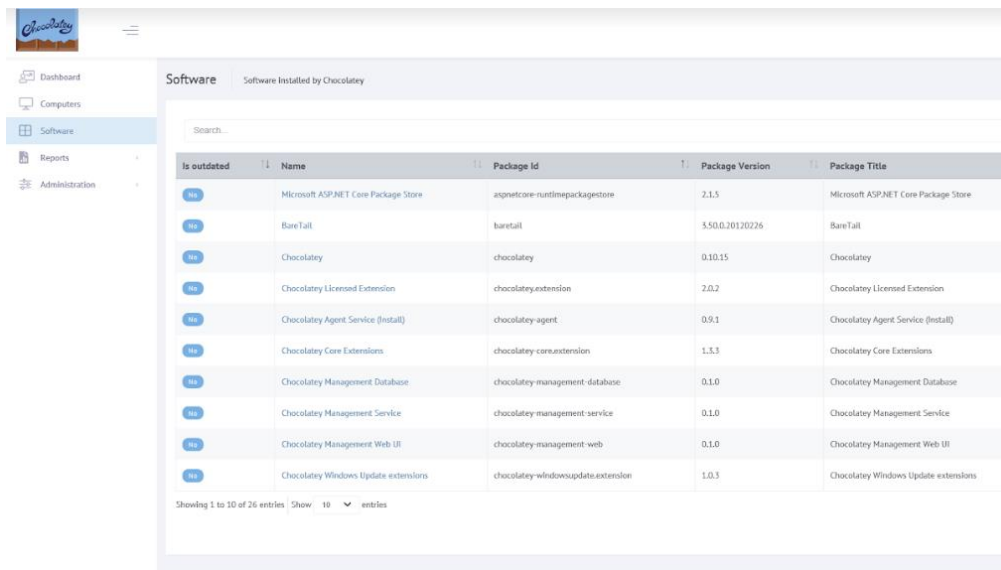
monitoring / alerting / statistics ?

- ...
-
-
- PS> iwr -Uri "\$url\$params" `
- -Method \$restMethod `
- -ContentType "application/json" `
- -Body \$logObject

...

monitoring / alerting / statistics ?

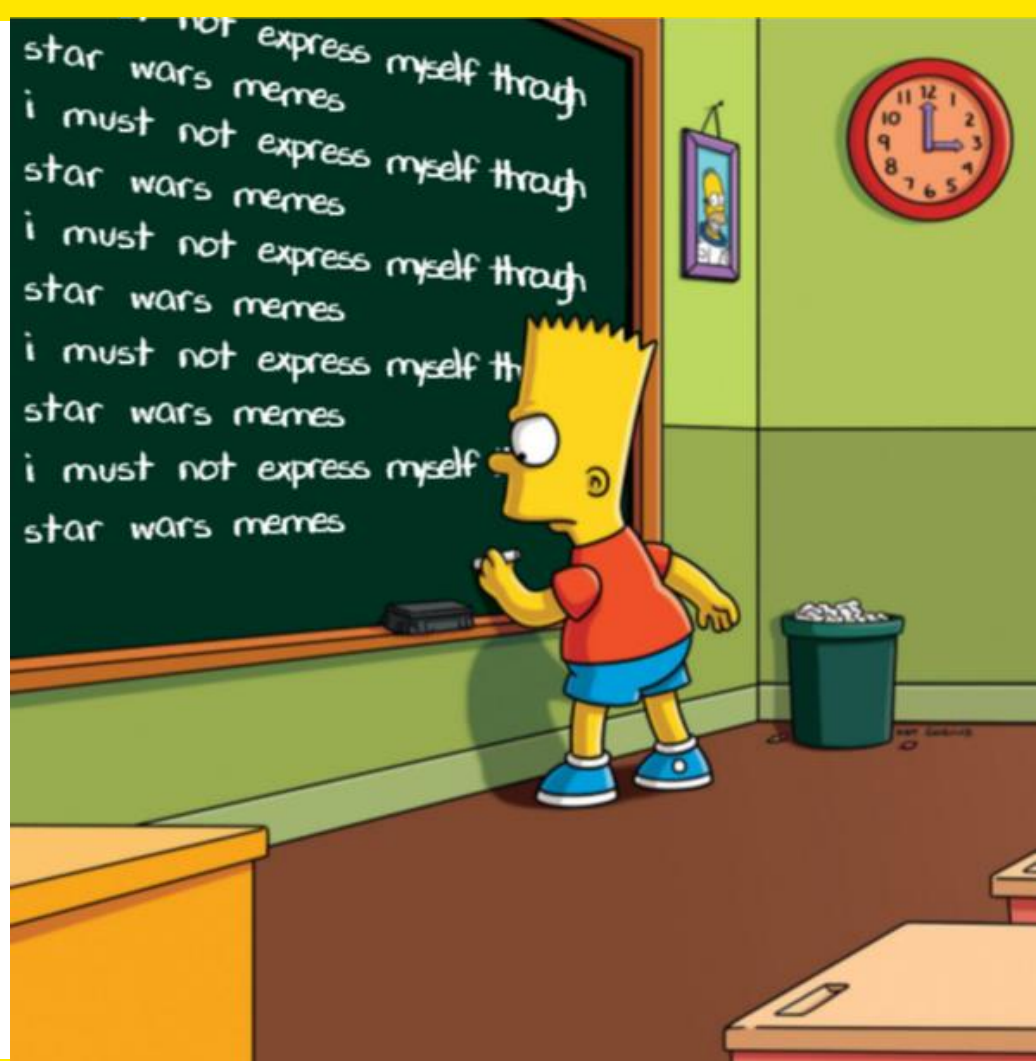
- DIY – PowerShell logfile analysis
- DIY – PoSh => ElasticSearch + Kibana
- C4B – Chocolatey Central Management
- ... ?



The screenshot displays the Chocolatey Central Management (CCM) web interface. On the left is a navigation sidebar with links to Dashboard, Computers, Software (selected), Reports, and Administration. The main content area is titled 'Software' and 'Software Installed by Chocolatey'. It features a search bar and a table of installed packages. The table has columns for 'Is outdated', 'Name', 'Package Id', 'Package Version', and 'Package Title'. Below the table, it indicates 'Showing 1 to 10 of 26 entries'.

Is outdated	Name	Package Id	Package Version	Package Title
No	Microsoft ASP.NET Core Package Store	aspnetcore-runtimepackagestore	2.1.5	Microsoft ASP.NET Core Package Store
No	BareTail	baretail	3.50.0.10120226	BareTail
No	Chocolatey	chocolatey	0.10.15	Chocolatey
No	Chocolatey Licensed Extension	chocolatey.licensed.extension	2.0.2	Chocolatey Licensed Extension
No	Chocolatey Agent Service (Install)	chocolatey-agent	0.9.1	Chocolatey Agent Service (Install)
No	Chocolatey Core Extensions	chocolatey-core.extensions	1.3.3	Chocolatey Core Extensions
No	Chocolatey Management Database	chocolatey-management-database	0.1.0	Chocolatey Management Database
No	Chocolatey Management Service	chocolatey-management-service	0.1.0	Chocolatey Management Service
No	Chocolatey Management Web UI	chocolatey-management-web	0.1.0	Chocolatey Management Web UI
No	Chocolatey Windows Update extensions	chocolatey-windowsupdate.extension	1.0.3	Chocolatey Windows Update extensions

the takeaway



the takeaway







conclusion

- golden images need a lot of manual work
- „hope is no strategy.“ (don't rely on the availability of remote resources)
- PowerShell is the tool of choice for SW automation
- Chocolatey is sweet
- Boxstarter is awesome



more...

let's talk!

I'm on the internets!

github.com/mwallner

[@schusterfredl](https://twitter.com/schusterfredl)

chocolatey@mwallner.net



404

this is not the slide
you are looking for