

Ansible для управления конфигурацией Windows

Алексей Чернов



Обо мне

- Занимаюсь всем, что связано с вебом примерно с 2004

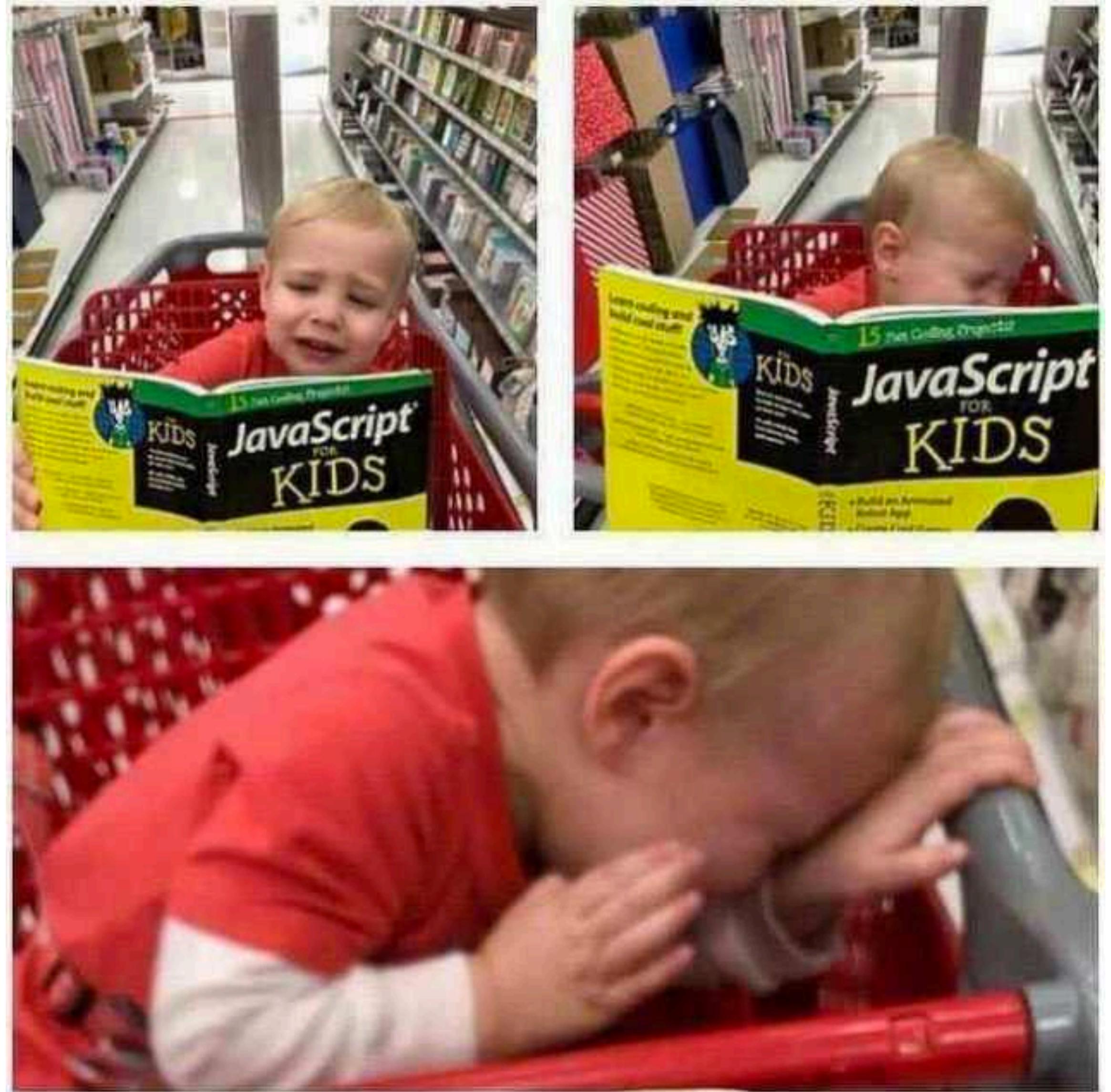
Обо мне

- Занимаюсь всем, что как-то связано с вебом примерно с 2004
- Perl, PHP, немного RoR



Обо мне

- Занимаюсь всем, что как-то связано с вебом примерно с 2004
- Perl, PHP, немного RoR
- JavaScript, Автотесты, Автоматизация



Предыстория

Предыстория

- Два года назад оказался в разработке продуктов под Windows

Предыстория

- Два года назад оказался в разработке продуктов под Windows
- Непривычно низкий уровень автоматизации деплоя

Предыстория

- Два года назад оказался в разработке продуктов под Windows
- Непривычно низкий уровень автоматизации деплоя
- Файлы деплоя приводят файлам автотестов



Текущие условия

Availability for the Always-On Enterprise

[Veeam Hyper-Availability Platform](#)

Virtual Workloads (VMware and Hyper-V)

[Veeam Availability Suite](#)

[Veeam Backup & Replication](#)

[Veeam Availability for Nutanix AHV](#)

Cloud and Physical Workloads

[Veeam Agent for Microsoft Windows](#)

[Veeam Agent for Linux](#)

[Veeam Agents for IBM AIX and Oracle Solaris](#)

[Veeam Availability for AWS](#)

[Veeam Backup for Microsoft Office 365](#)

Disaster Recovery Orchestration

[Veeam Availability Orchestrator](#)

[ALL PRODUCTS](#)

Small and Medium Businesses

[Veeam Backup Essentials](#)

[Special Offer](#)

Cloud and Service Providers

[VCSP Program and Product Offerings](#)

[Veeam Availability Console](#)

Monitoring and Management

[Veeam ONE](#)

[Veeam Management Pack for System Center](#)

Free Tools

[Veeam Backup & Replication Community Edition](#)

[Veeam Agent for Microsoft Windows FREE](#)

[Veeam Agent for Linux FREE](#)

[Veeam Backup for Microsoft Office 365 Community Edition](#)

[Veeam Powered Network](#)

[Other Free Tools](#)

Задачи автоматизации деплоя

- Ежедневно производится много билдов приложений

Задачи автоматизации деплоя

- Ежедневно производится много билдов приложений
- Большинство продуктов под Windows

Задачи автоматизации деплоя

- Ежедневно производится много билдов приложений
- Большинство продуктов под Windows
- Большая инфраструктура

Проблемы

Проблемы

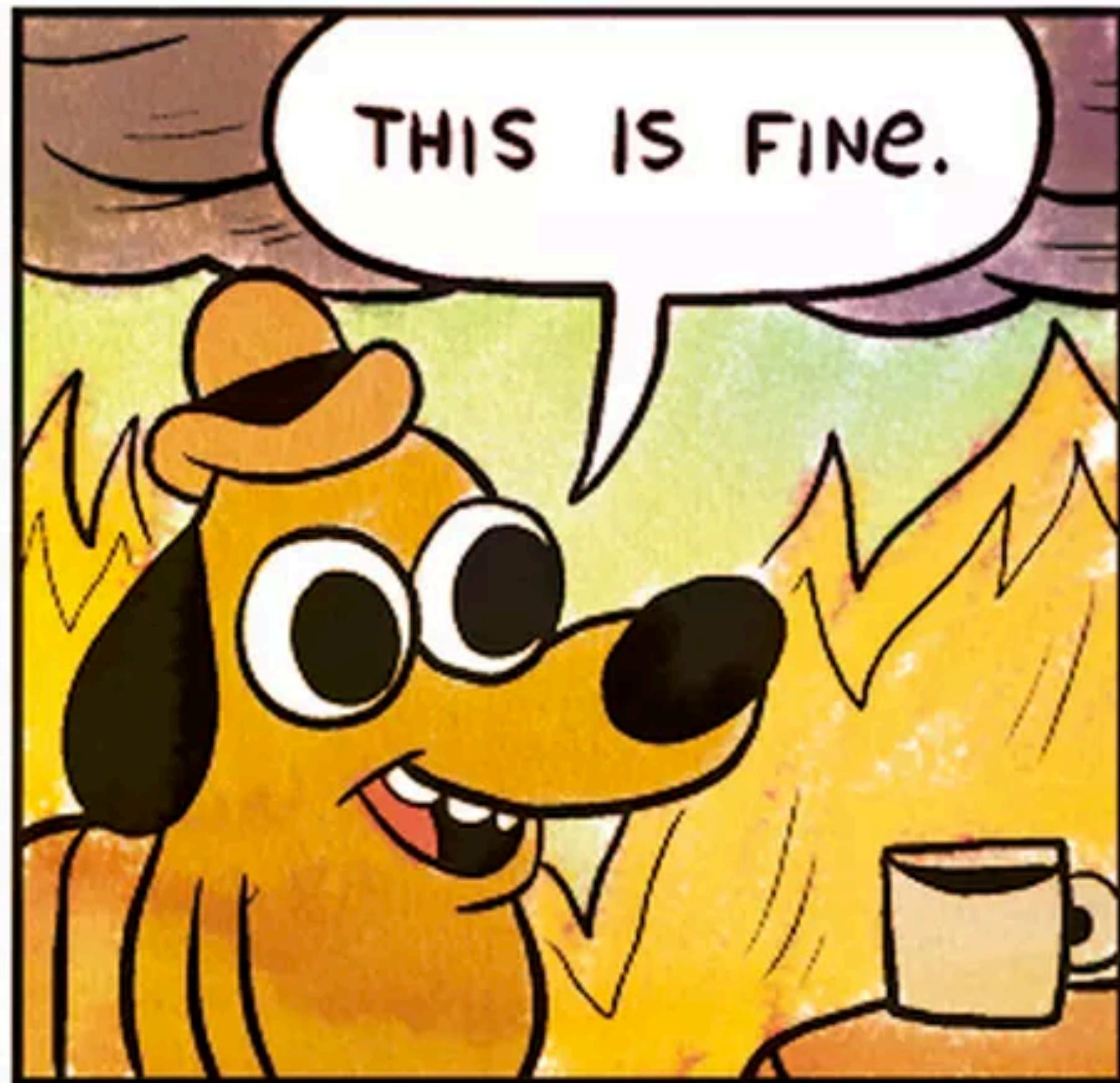
- Переусложнённая реализация Powershell скриптов

Проблемы

- Переусложнённая реализация Powershell скриптов
- Затруднён сбор результатов деплоя

Проблемы

- Переусложнённая реализация Powershell скриптов
- Затруднён сбор результатов деплоя
- Онбординг новых членов команды



Подбор решения

Требования

- Единая кодовая база

Требования

- Единая кодовая база
- Все знают как это работает

Требования

- Единая кодовая база
- Все должны знать, как это работает
- Возможность запуска по кнопке

Попытки решения

Попытки решения

- Объединить Powershell-скрипты в один репозиторий

Попытки решения

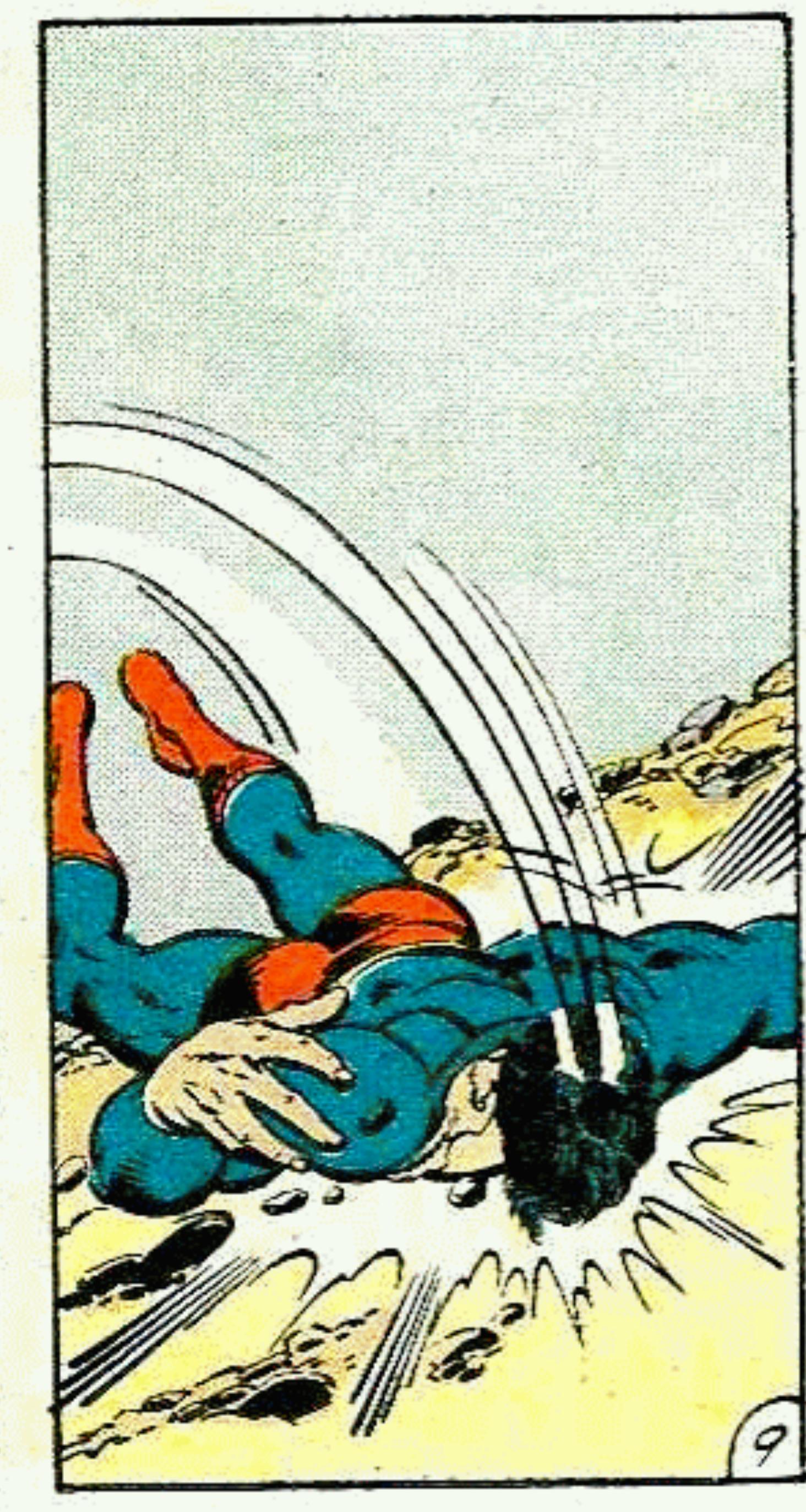
- Объединить Powershell-скрипты в один репозиторий
Fail: Уtkнулись в холивары и форки

Попытки решения

- Объединить Powershell-скрипты в один репозиторий
Fail: Уtkнулись в холивары и форки
- Перенести скрипты на CI-сервер (TeamCity)

Попытки решения

- Объединить Powershell-скрипты в один репозиторий
Fail: Уtkнулись в холивары и форки
- Перенести скрипты на CI-сервер (TeamCity)
Fail: проблемы версионирования и совместной работы



 **addCloudJob.ps1**

 **addCloudProvider.ps1**

 **addCloudReplica.ps1**

 **addFailoverPlan.ps1**

 **addLocalJob.ps1**

 **resoreVM.ps1**

📄 addCloudJob.ps1

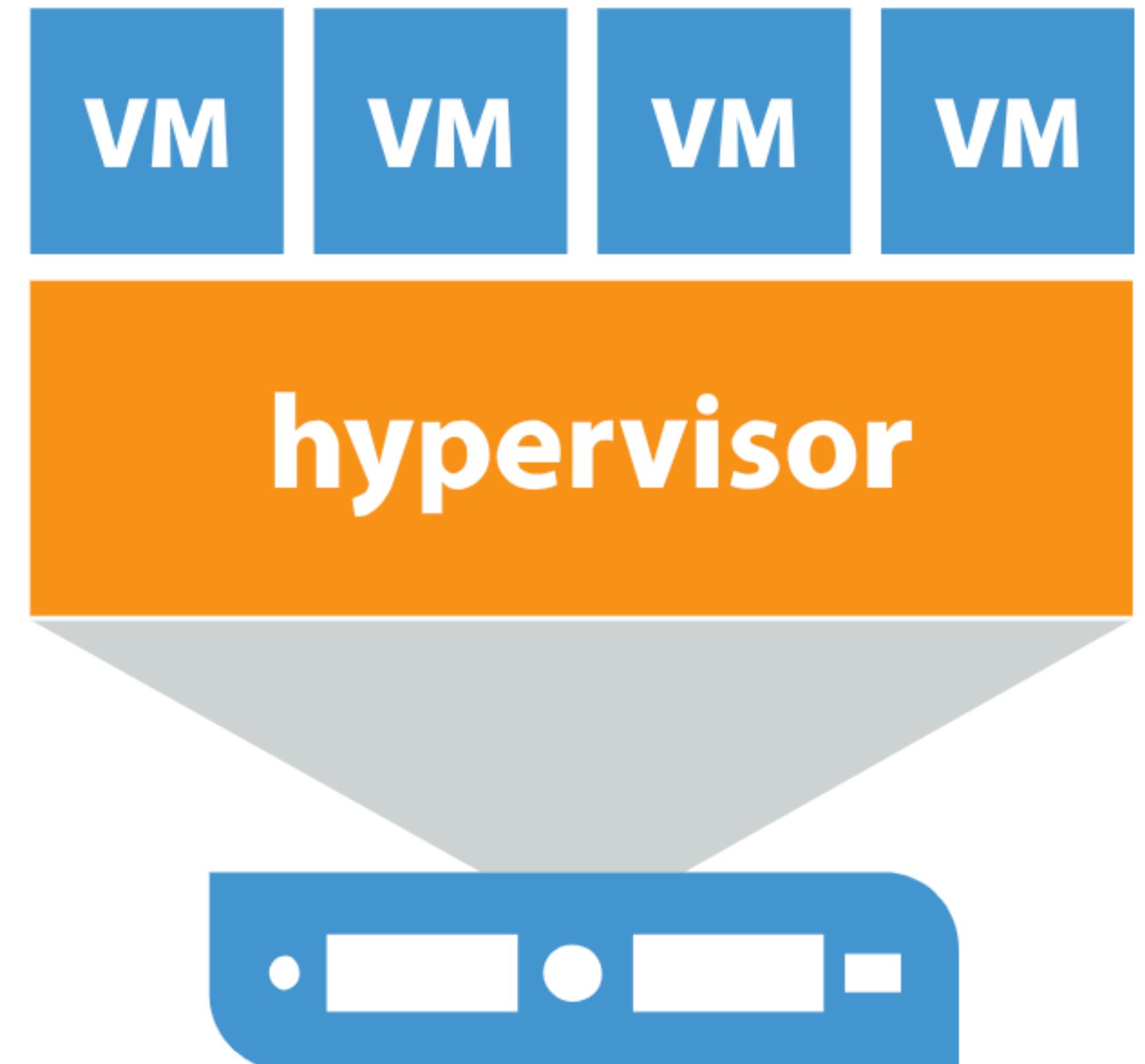
📄 addCloudProvider.ps1

📄 addCloudReplica.ps1

📄 addFailoverPlan.ps1

📄 addLocalJob.ps1

📄 resoreVM.ps1



📄 addCloudJob.ps1

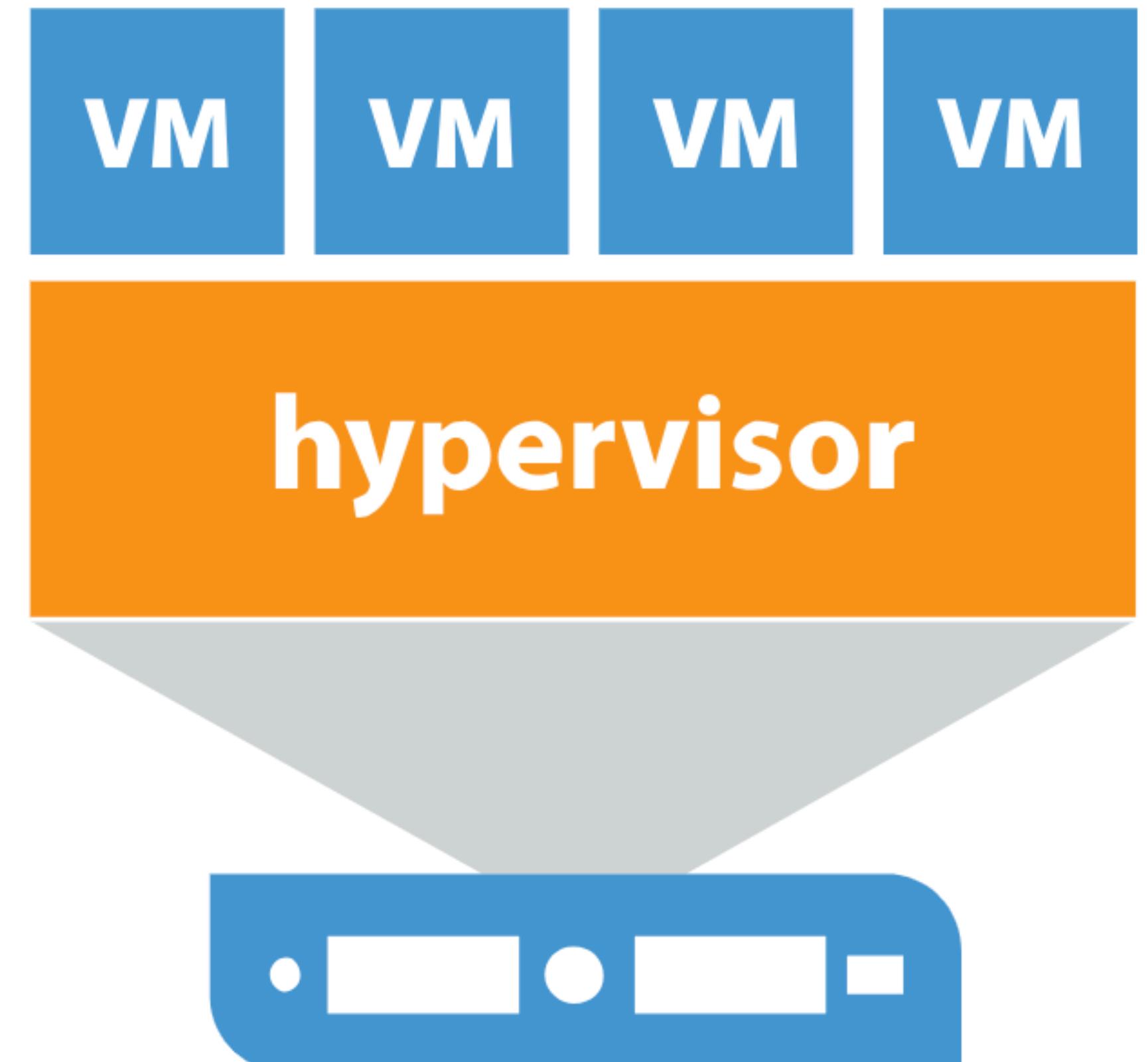
📄 addCloudProvider.ps1

📄 addCloudReplica.ps1

📄 addFailoverPlan.ps1

📄 addLocalJob.ps1

📄 resoreVM.ps1



Не хватает описания инфраструктуры

Infrastructure as Code

Лучшие практики из разработки

Лучшие практики из разработки

- Version Control

Лучшие практики из разработки

- Version Control
- Code Review

Лучшие практики из разработки

- Version Control
- Code Review
- Continuous Integration

Configuration Management Tools

◆	Language ◆	License ◆	Mutual auth ◆	Encrypts ◆	Verify mode ◆	Agent-less ◆	Have a GUI ◆	First release ◆	Latest stable release ◆
Ansible	Python	GPLv3+	Yes ^[1]	Yes ^[2]	Yes	Yes	Yes ^[3]	2012-03-08	2018-01-31 2.4.3.0 ^{[4][5]}
Bcfg2	Python	BSD 2-clause ^[6]	Yes ^[7]	Yes ^[8]	Yes ^[9]	No	Yes ^[10]	2004-08-11 ^[11]	2015-06-11 1.3.6 ^[11]
Capistrano	Ruby	MIT License		Yes ^[2]		Yes ^[2]	No	2005	2017-03-08 4.4.2
cdist	Python	GPLv3+	Yes ^[1]	Yes ^[2]		Yes		2010	2018-04-20 4.8.4 ^[12]
Chef	Ruby, Erlang	Apache 2.0	Yes ^[13]	Yes ^[14]	Yes ^{[15][16]}	No	Yes	2009-01-15 0.5.0	2018-09-25 14.5.33 (client), ^[17] 2018-02-13 12.17.33 (server) ^[18]
CFEngine	C ^[19]	GPLv3 ^[20]	Yes ^[1]	Yes ^[21]	Yes ^{[22][23]}	No		1993	2018-11-23 3.13.0 ^[24] , 2018-11-30 3.12.1 ^[25] , 2018-11-16 3.10.5 ^[26]
ISconf	Python	GPL ^[27]	Yes ^[28]	No ^[29]				1998	2006-08-13 4.2.8.233
Juju	Python, Go ^[30]	Affero General Public License	Yes ^[1]	Yes ^[8]	No	No	Yes ^[31]	2010-09-17 ^[32]	2017-03-16 1.2.1 ^[33]

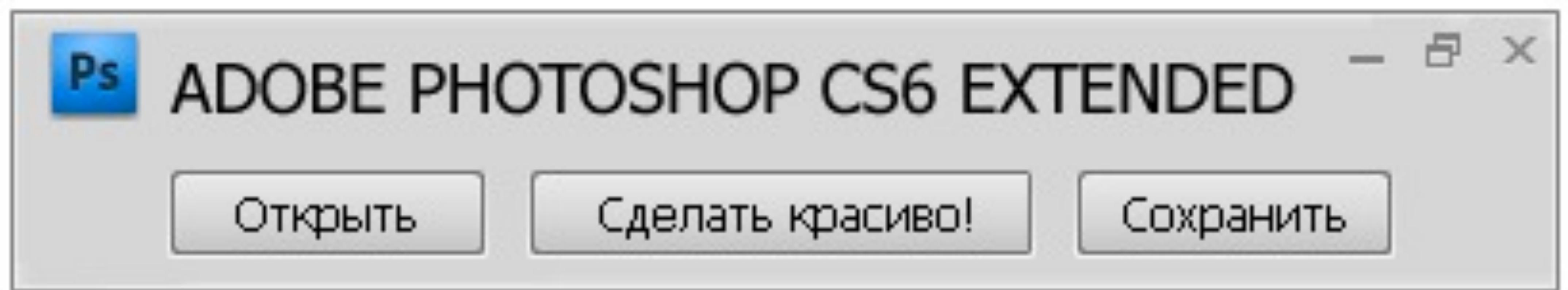
	Language			Public License						17 ^[32]	1.2.1 ^[33]
		Local ConFiGuration system (LCFG)	Perl	GPL	Partial ^[34]	Partial ^[35]				1994	Weekly Releases
Ansible	Python	NOC	Python	BSD License 2.0	Yes ^[1]	Yes ^[2]	Yes	Yes	Yes	2012-03-08	2015-05-20 15.05.1 ^[36]
Bcfg2	Python	OCS Inventory NG with GLPI	Perl, PHP, C++	GPL	No ^[37]	Yes ^[8]		No		2003	2014-07-13 ^[38]
Capistrano	Ruby	Open pc server integration (Opsi)	Python, Java	GPL	No	Yes ^[8]		No		2004	2013-03-01 4.0.3
cdist	Python	PIKT	C	GPLv2+ ^[39]	Yes ^[40]	Yes ^[41]		No		1998 ^[42]	2007-09-10 1.19.0
Chef	Ruby, Erlang	Puppet	C++ & Clojure from 4.0, ^[43] Ruby before then	Apache from 2.7.0, GPL before then	Yes ^[44]	Yes ^[8]	Yes ^{[45][46]}	No	Yes ^[47]	2005-08-30 ^[48]	2017-05-11 4.10.1 ^[49]
CFEngine	C ^[19]	Quattor	Perl, Python	Apache 2.0 ^{[50][51]}	Yes ^[52]	Yes ^[53]				2005-04-01 ^[54]	2017-10-31 17.8.0 ^[55]
		Radmind	C	BSD ^[56]	Yes ^[57]	Yes ^[58]		No		2002-03-26 ^[59]	2008-10-08 1.13.0 ^[60]
		Rex	Perl	Apache	Yes ^[1]	Yes ^[2]		Yes		2010-11-05 0.9.0 ^[61]	2017-12-03 1.6.0 ^[62]
		Rudder	C and Scala	GPLv3 and Apache 2.0 ^[63]	Yes ^[1]	Yes ^[8]	Yes ^{[64][65]}	No	Yes	2011-10-31	2018-04-19 4.3.0 ^[66]
ISconf	Python	SmartFrog	Java	LGPL	Yes ^[67]	Yes ^[67]		No		2004-02-11	2009-01-26 3.16.004 ^{[68][69]}
Juju	Python, Go ^[30]	Salt ^[70]	Python ^[71]	Apache 2.0 ^[72]	Yes ^[73]	Yes ^[73]	Yes	Both ^{[74][75]}	Yes ^{[76][77]}	2011-03-17	2017-09-25 2018-03-09 ^[79]



ANSIBLE

Почему Ansible

**Чего мы на самом деле хотим когда говорим
об автоматизации?**



Идемпотентность

Идемпотентность

```
1 $url = "http://buildserver/build.msi"  
2 $output = "$PSScriptRoot\build.msi"  
3 Invoke-WebRequest -Uri $url -OutFile $output  
4
```

```
1 └ - name: Download build  
2     hosts: all  
3     tasks:  
4         └ - name: Download installer  
5             win_get_url:  
6                 url: "http://buildserver/build.msi"  
7                 dest: "build.msi"  
8
```

Идемпотентность

```
171  
172     $module.Result.status_code = 200  
173     $module.Result.changed = $true  
174     $module.Result.msg = 'OK'  
175     $module.Result.dest = $dest  
176     $module.Result.elapsed = ((Get-Date) - $module_start).TotalSeconds  
177 }  
178  
179 $module.Result.dest = $dest  
180 $module.Result.elapsed = 0  
181 $module.Result.url = $url  
182  
183 if (-not $use_proxy -and ($proxy_url -or $proxy_username -or $proxy_password)) {  
184     $module.Warn("Not using a proxy on request, however a 'proxy_url' or 'proxy_username' parameter was provided")  
185 }  
186  
187 $proxy = $null  
188 if ($proxy_url) {  
189     $proxy = New-Object System.Net.WebProxy($proxy_url, $true)  
190     if ($proxy_username -and $proxy_password) {  
191         $proxy_credential = New-Object System.Net.NetworkCredential($proxy_username, $proxy_password)  
192         $proxy.Credentials = $proxy_credential  
193     }  
194 }
```



```
1 ┌ - name: Download build  
2   hosts: all  
3   tasks:  
4     ┌ - name: Download installer  
5       win_get_url:  
6         url: "http://buildserver/build.msi"  
7         dest: "build.msi"  
8         force: no  
9
```

Идемпотентность

257 строк кода

Идемпотентность везде

- Проверить, существует ли виртуальная машина?

- Проверить, существует ли виртуальная машина?
- Установлен ли msi пакет?

- Проверить, существует ли виртуальная машина?
- Установлен ли msi пакет?
- Надо ли заново качать билд артефакты?

Почему Ansible

- Идемпотентность
- Agent-less architecture

Почему Ansible

- Идемпотентность
- Agent-less architecture
- YAML - декларативный и human-readable

Почему Ansible

- Идемпотентность
- Agent-less architecture
- YAML - декларативный и human-readable
- Cross-platform*

Минусы

Минусы Ansible

- Для запуска плейбуков нужен Linux

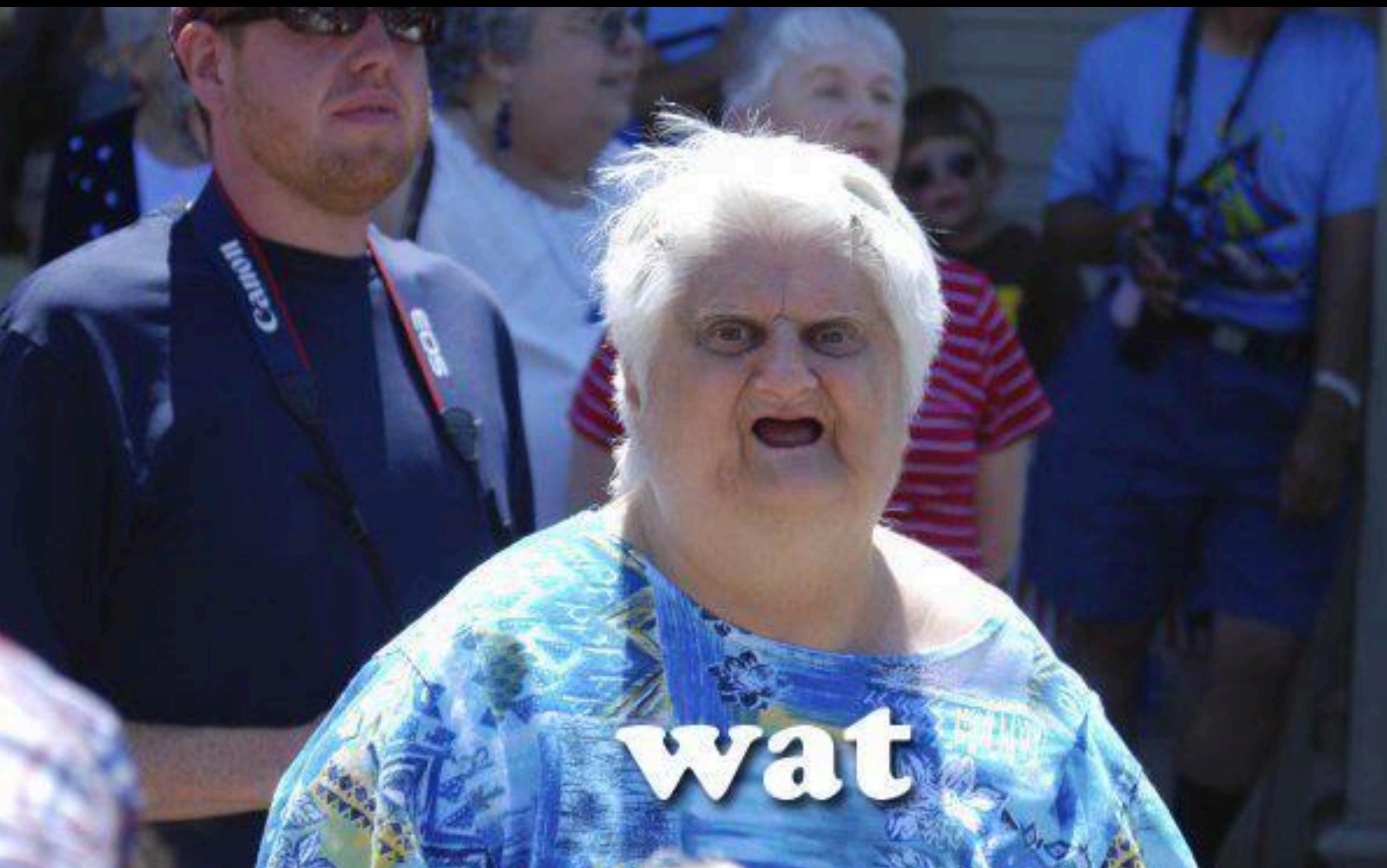
Минусы Ansible

- Для запуска плейбуков нужен Linux
Теперь есть WSL

Минусы Ansible

- Для запуска нужен Linux // теперь есть WSL
- Иногда сложно отлаживать

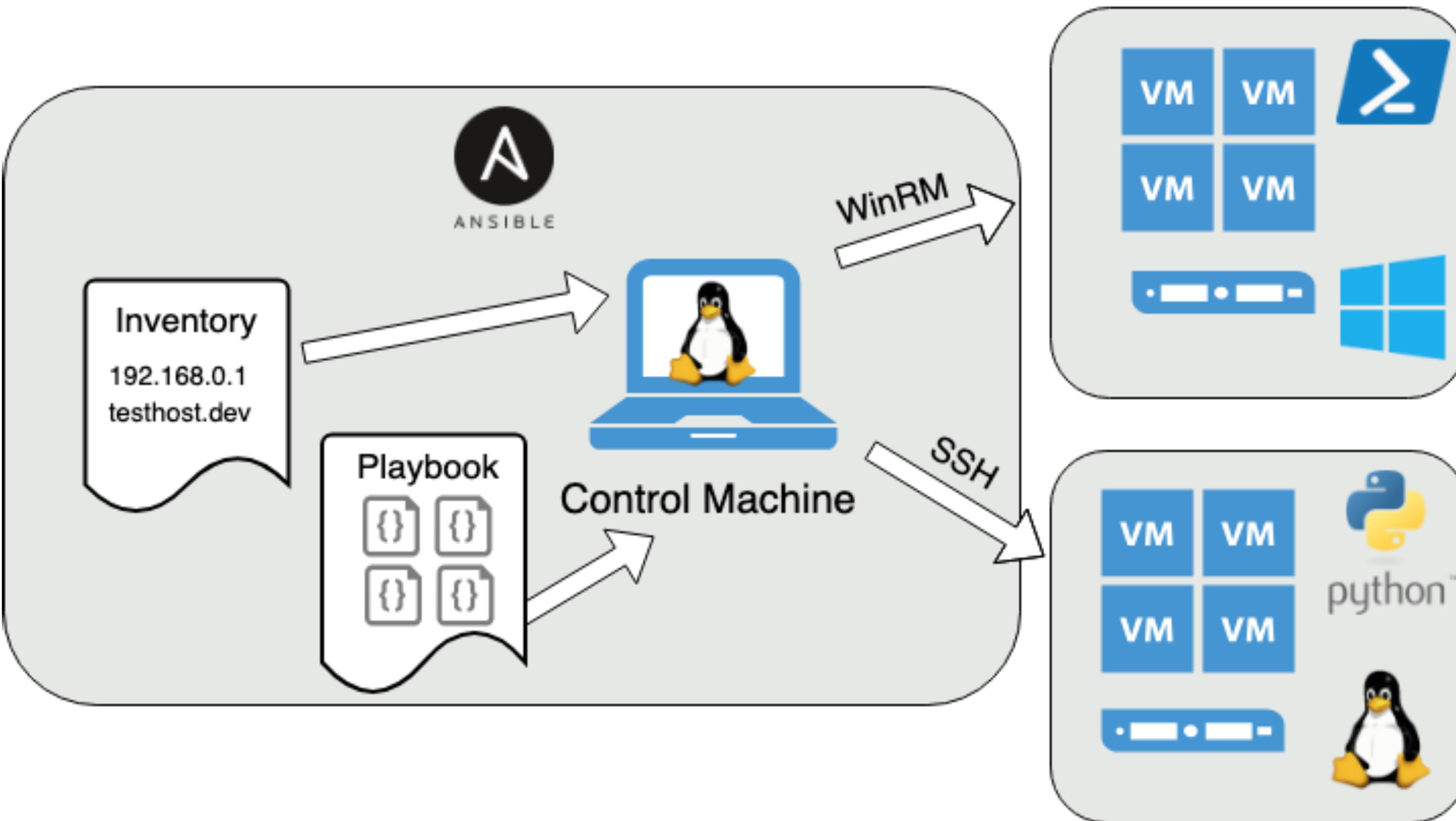
```
task path: /  
Using module  
<172.24.167.  
EXEC (via pi  
fatal: [spbo  
  "changed  
  "module_  
    ~\r\n  
    ~\r\nMi  
in if statem  
:21 char:30\  
\n+      if s  
        MOD  
r\n+  
  expression  
Missing clos  
  
nd try again  
~~~~~  
on\r\n \r\nT  
pt block, or  
ion: (: ) [],  
  "module_  
  "msg": "  
  "rc": 1  
}
```



```
) :\r\n+  
ssing '(' aft  
statement.\r\nAt line:21 c  
line:23 char  
\nAt line:23  
en 'imp.PY_SO  
  
the reported  
~~~~~  
rrorId : Pars  
hand name,\r\n  : Inval
```

Как это работает?

Как это работает?



Что нужно

- Python и pip

Что нужно

- Python и pip
- pip install ansible

Что нужно

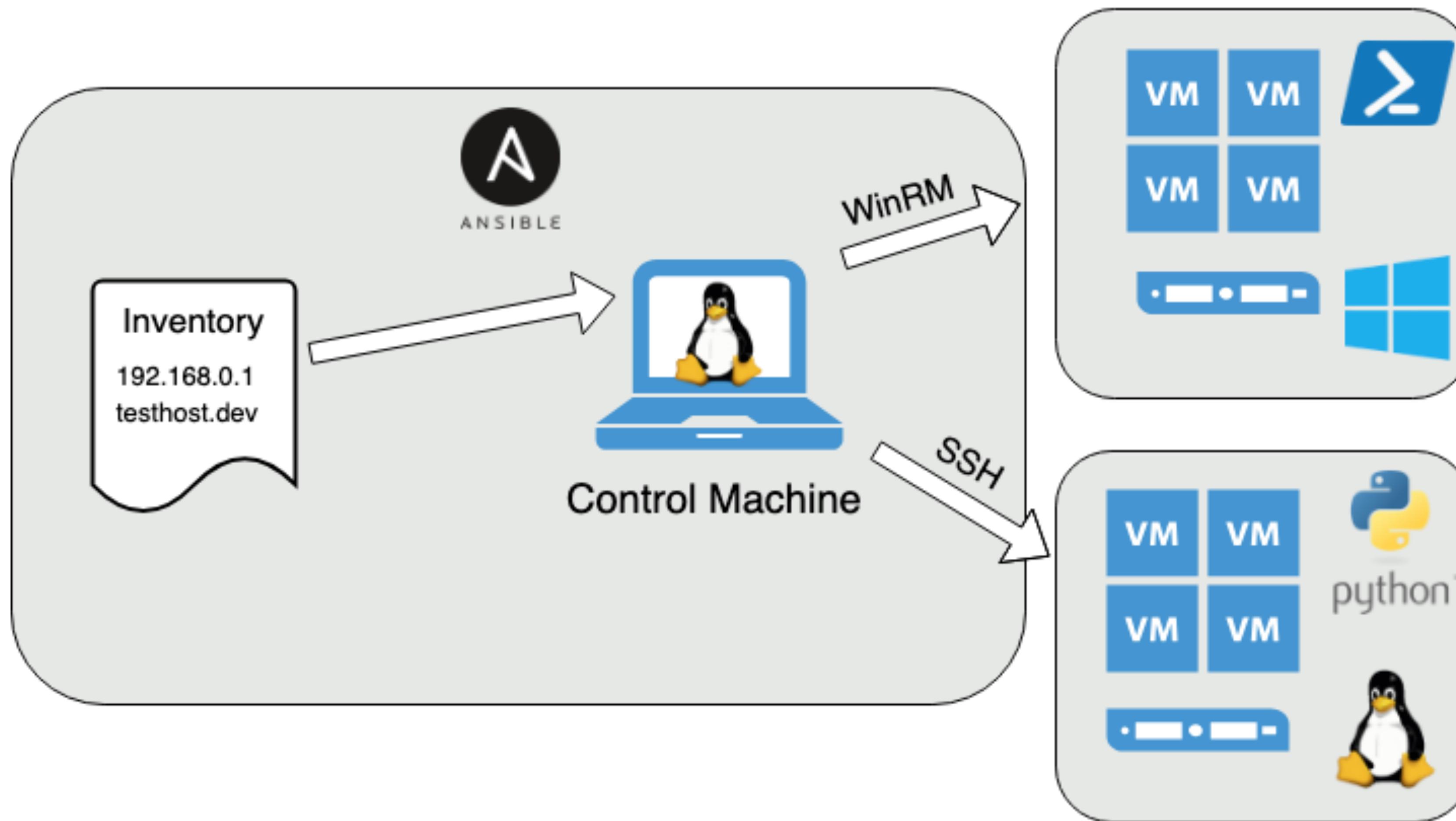
- Python и pip
- pip install ansible
- pip install pywinrm[credssp]

Что нужно

- Python и pip
- pip install ansible
- pip install pywinrm[credssp]
- На управляемых хостах: ConfigureRemotingForAnsible.ps1 -EnableCredSSP

Основные концепции

Inventory



! inventory.yml ●

! dotnet-app.yml

! install-aws-cli.yml

```
1 all:
2   children:
3     webservers:
4       hosts:
5         spbdotnet-test-host.dev.local:
6     dbservers:
7       hosts:
8         spbdotnet-test-host.dev.local:
9   vars:
10    ansible_connection: winrm
11    ansible_winrm_transport: credssp
12    ansible_winrm_server_cert_validation: ignore
13    ansible_user: administrator
14    ansible_password: 123
15
```

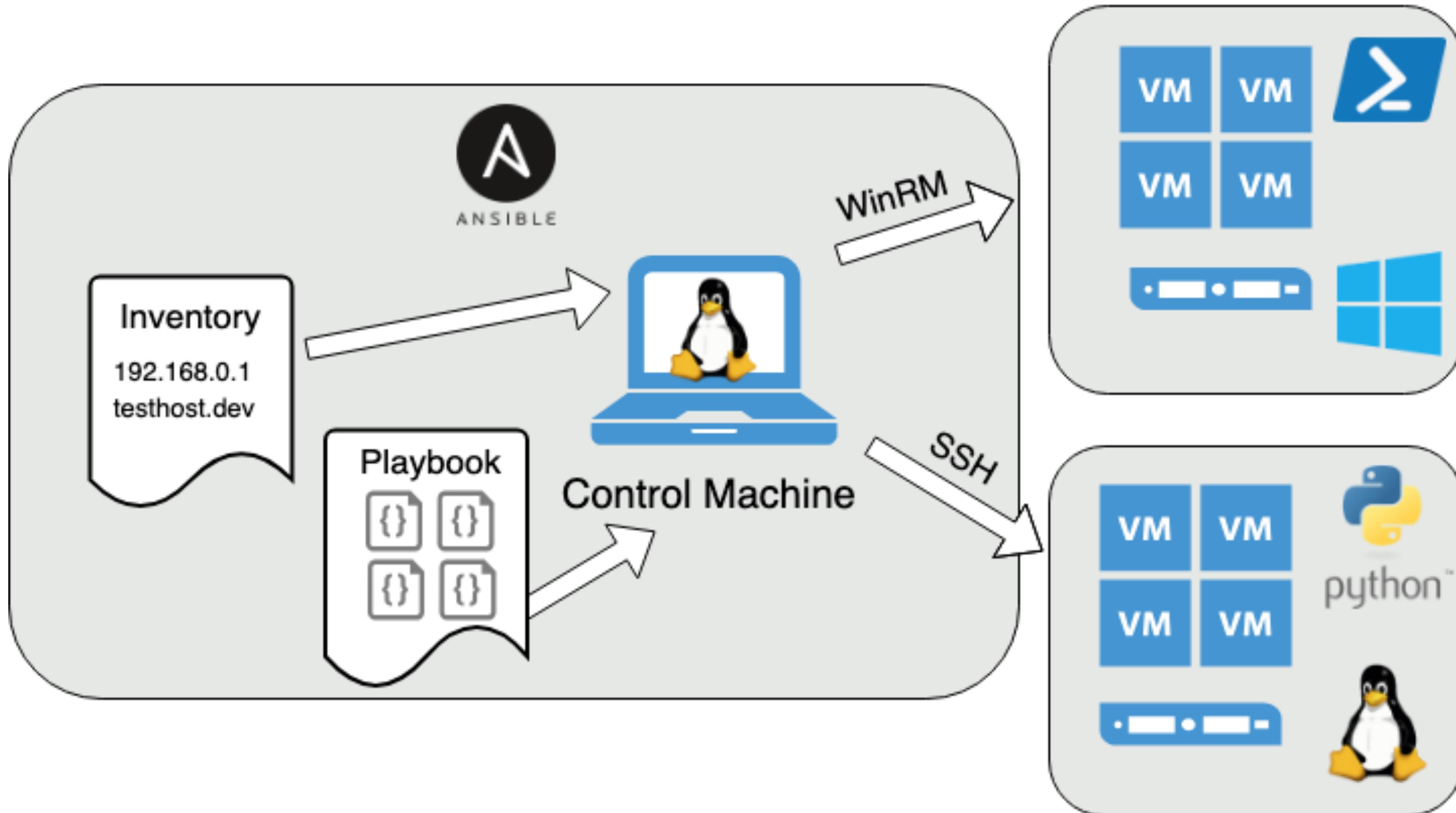
Модули

```
root@6af48d6254d1:/# ansible all -i inventory.yml -m win_ping
```

```
spbdotnet-test-host | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

```
7 $spec = @{
8     options = @{
9         data = @{
10            type = "str";
11            default = "pong"
12        }
13        supports_check_mode = $true
14    }
15
16    $module = [Ansible.Basic.AnsibleModule]::Create($args, $spec)
17    $data = $module.Params.data
18
19
20    if ($data -eq "crash") {
21        throw "boom"
22    }
23
24    $module.Result.ping = $data
25
26    $module.ExitJson()
```

Playbook



```
1 - name: Install AWS CLI
2   hosts: all
3   tasks:
4     - name: Ensure target directory exists
5       win_file:
6         path: c:\downloads
7         state: directory
8     - name: Download installer
9       win_get_url:
10        url: https://s3.amazonaws.com/aws-cli/AWSCLI32PY3.msi
11        dest: c:\downloads\awscli.msi"
12        force: no
13     - name: Install AWS CLI
14       win_package:
15         path: c:\downloads\awscli.msi"
16         state: present
```

```
1  - name: Install AWS CLI
2    hosts: all
3    tasks:
4      - name: Ensure target directory exists
5        win_file:
6          path: c:\downloads
7          state: directory
8      - name: Download installer
9        win_get_url:
10         url: https://s3.amazonaws.com/aws-cli/AWSCLI32PY3.msi
11         dest: c:\downloads\awscli.msi"
12         force: no
13      - name: Install AWS CLI
14        win_package:
15          path: c:\downloads\awscli.msi"
16          state: present
```

```
[root@6af48d6254d1:/# ansible-playbook install-aws-cli.yml -i inventory.yml

PLAY [Install AWS CLI] *****
TASK [Gathering Facts] *****
ok: [spbdotnet-test-host]

TASK [Ensure target directory exists] *****
changed: [spbdotnet-test-host]

TASK [Download installer] *****
changed: [spbdotnet-test-host]

TASK [Install AWS CLI] *****
changed: [spbdotnet-test-host]

PLAY RECAP *****
spbdotnet-test-host : ok=4      changed=3      unreachable=0      failed=0
```

Идемпотентность

```
1 - name: Install AWS CLI
2   hosts: all
3   tasks:
4     - name: Ensure target directory exists
5       win_file:
6         path: c:\downloads
7         state: directory
8     - name: Download installer
9       win_get_url:
10      url: https://s3.amazonaws.com/aws-cli/AWSCLI32PY3.msi
11      dest: c:\downloads\awscli.msi"
12      force: no
13     - name: Install AWS CLI
14       win_package:
15         path: c:\downloads\awscli.msi"
16         state: present
```

```
[root@6af48d6254d1:/# ansible-playbook install-aws-cli.yml -i inventory.yml

PLAY [Install AWS CLI] ****
TASK [Gathering Facts] ****
ok: [spbdotnet-test-host]

TASK [Ensure target directory exists] ****
ok: [spbdotnet-test-host]

TASK [Download installer] ****
ok: [spbdotnet-test-host]

TASK [Install AWS CLI] ****
ok: [spbdotnet-test-host]

PLAY RECAP ****
spbdotnet-test-host : ok=4      changed=0      unreachable=0      failed=0
```

Деплой приложения

Подготовка

```
1 ┌ - name: Setup App
2   ┌ hosts: webservers
3   ┌ tasks:
4   ┌   ┌ - name: Install IIS
5   ┌   ┌   win_feature:
6   ┌   ┌     name:
7   ┌   ┌       - Web-Server
8   ┌   ┌       - Web-Common-Http
9   ┌   include_sub_features: True
10  ┌   include_management_tools: True
11  ┌   state: present
12  ┌   register: win_feature
13  ┌   ┌ - name: reboot if installing Web-Server feature requires it
14  ┌   ┌   win_reboot:
15  ┌   ┌     when: win_feature.reboot_required
16
```

```
1 ┌ - name: Setup App
2   hosts: webservers
3   tasks:
4     - name: Install IIS
5       win_feature:
6         name:
7           - Web-Server
8           - Web-Common-Http
9         include_sub_features: True
10        include_management_tools: True
11        state: present
12       register: win_feature
13     - name: reboot if installing Web-Server feature requires it
14       win_reboot:
15       when: win_feature.reboot_required
16
```

```
14  
15 └ - name: SQL Server  
16   hosts: dbservers  
17   tasks:  
18     - name: Install MS SQL Server 2014  
19       win_chocolatey:  
20         name: mssqlserver2014express  
21         state: present
```



<https://chocolatey.org/>

The screenshot shows the homepage of the Chocolatey website. At the top, there's a dark header bar with a light orange background. On the left is the Chocolatey logo, which consists of a brown circle with a white dot and the word "chocolatey" in a brown, sans-serif font. To the right of the logo is a notice: "Notices: This section not yet converted to new layout. Download stats are rolling back out." Below the header is a navigation bar with links for Home, About, Compare, Packages (which is highlighted in blue), Upload, Docs, Forum*, and Shop*. To the right of the navigation bar is a search bar with the placeholder "Search Packages" and a magnifying glass icon. At the far right of the header are "Login" and "Signup" buttons.

Your use of the packages on this site means you understand they are not supported or guaranteed in any way. Due to the nature of a public repository and unreliability due to distribution rights, these packages should not be used as is for organizational purposes either. [Learn more](#).

There are 6322 community maintained packages
Displaying results 1 - 30.

[Subscribe to updates](#)
Or subscribe by [email](#)

Normal View

Stable Only

Sort By [Popularity](#)

Запуск приложений

Запуск приложений



```
23      - name: Deploy binaries
24      hosts: webservers
25      vars:
26          | myapp_workdir: C:\myapp
27      tasks:
28          - name: Remove Service if exist
29              | win_service:
30                  |     | name: MyAppService
31                  |     | state: absent
32                  |     | path: "{{ myapp_workdir }}\\MyAppService.exe"
```

```
49      - name: Delete old files
50          win_file:
51              path: "{{ myapp_workdir }}\\"
52              state: absent
53
54      - name: Copy artifacts to remote machine
55          win_copy:
56              src: "{{ myapp_artifacts }}"
57              dest: "{{ myapp_workdir }}\\"
58
59      - name: Unzip build artifacts
60          win_unzip:
61              src: "{{ myapp_workdir }}\\MyAppService.zip"
62              dest: "{{ myapp_workdir }}"
```

```
67
68 └─ - name: Register and start the service
69   └─ win_service:
70     ┌─ name: ReporterService
71     ┌─ start_mode: auto
72     ┌─ state: started
73     ┌─ path: "{{ myapp_workdir }}\\MyAppService.exe"
74
```

Готово!?

Идемпотентность!

```
29      - name: Get artifacts checksum
30          stat:
31              path: "{{ myapp_artifacts }}"
32          delegate_to: localhost
33          register: myapp_artifacts_stat
34
35      - name: Get remote artifacts checksum
36          win_stat:
37              path: "{{ myapp_workdir }}\\MyAppService.zip"
38          register: myapp_remote_artifacts_stat
39
```

```
29      - name: Get artifacts checksum
30          stat:
31              path: "{{ myapp_artifacts }}"
32          delegate_to: localhost
33          register: myapp_artifacts_stat
34
35      - name: Get remote artifacts checksum
36          win_stat:
37              path: "{{ myapp_workdir }}\\MyAppService.zip"
38          register: myapp_remote_artifacts_stat
39
```

```
59
40     - name: Stop play if checksums match
41         meta: end_play
42     when:
43         - myapp_artifacts_stat.stat.checksum is defined
44         - myapp_remote_artifacts_stat.stat.checksum is defined
45         - myapp_artifacts_stat.stat.checksum == myapp_remote_
46
```

IIS Web Application

```
40  - name: Ensure that the WebApp application exists
41    win_iis_webapplication:
42      name: WebApp
43      physical_path: c:\webapp
44      site: Default Web Site
45
```

Переиспользование кода

Роли

Example project structure:

```
site.yml  
webservers.yml  
fooservers.yml  
roles/  
  common/  
  tasks/  
  handlers/  
  files/  
  templates/  
  vars/  
  defaults/  
  meta/  
webservers/  
  tasks/  
  defaults/  
  meta/
```

```
1 - name: Run App
2   hosts: webservers
3   tasks:
4     - name: "Install IIS"
5       include_role:
6         name: IIS
7     - name: Run My App
8       include_role:
9         name: MyAppService
10      vars:
11        myapp_artifacts: ./build.zip
12
```

Ansible Galaxy



GALAXY

 [Home](#) [Search](#) [Community](#) [Home](#)

Most Popular

 [System](#) [Development](#) [Networking](#) [Cloud](#) [Database](#) [Monitoring](#) [Packaging](#) [Playbook Bundles](#) [Security](#) [Web](#)

Больше гибкости?

What's covered in this section:

- Windows environment setup
- Create a Windows server in a VM
- Create an Ansible inventory
- Provisioning the environment
- Windows new module development
- Windows module utilities
- Windows playbook module testing
- Windows debugging
- Windows unit testing
- Windows integration testing
- Windows communication and development support

Continuous Integration



Проверка конфигов

- Syntax Check, Diff, Check mode

Проверка конфигов

- Syntax Check, Diff, Check mode
- ansible-lint

Проверка конфигов

```
[root@072884057d26:/work# ansible-lint dotnet-app.yml
[201] Trailing whitespace
dotnet-app.yml:15
    when: win_feature.reboot_required

[502] All tasks should be named
dotnet-app.yml:20
Task/Handler: win_chocolatey state=present name=mssqlserver2014express __file__=dotnet-app.yml __l
[...]
```

Проверка конфигов

- Syntax Check, Diff, Check mode
- ansible-lint
- Тесты

Проверка конфигов

- Syntax Check, Diff, Check mode
- ansible-lint
- Тесты - важно, если вы деплоите на продакшн

Запуск плейбуков

Run:

Custom script

Custom script: *

Enter build script content:

```
ansible-playbook reporter_vm.yml -i inventory.yml -vvvv
```



Labs

Overview

Change Log

Statistics

Current Problems

Investigations

Muted Problems

Flaky Tests 0

Slack

 Hide successful configurations

Basic configuration | ▾

Deploy ...

#248

Success | ▾

Changes (7) | ▾

14 hours ago (19m:49s)



Advanced Configuration | ▾

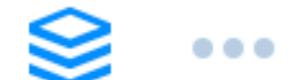
Deploy ...

#206

Success | ▾

No changes | ▾

15 hours ago (18m:59s)



Autotest Lab

<All branches>



Branches Overview History Change Log Issue Log Statistics Compatible Agents

Build chains: [?](#) <All build chains>

Showing 20 most recent build chains.

▼ ... Autotest Lab master #53 Success

Show details Group composite builds Group by projects

VeeamONE / Reporter V10 / Server / Build
develop
#61 Success |
Changes (1) |

... Autotest Lab
master
#53 Success |
Changes (2) |

VeeamONE / Reporter V10 / Web UI / Build
develop
#103 Success |
Changes (1) |

! #41 (07 Feb 19 18:45) | ▾

Overview Changes Build Log Parameters Dependencies Artifacts

Tree view | Tail



View: All messages

Console view

```
[18:45:02] ▶ Collecting changes in 2 VCS roots
[18:45:03] The build is removed from the queue to be prepared for the start
[18:45:03] Starting the build on the agent Linux Production 1
[18:45:03] Clearing temporary directory: /home/teamcity/buildAgent/temp/buildTmp
[18:45:03] ▶ Publishing internal artifacts (2s)
[18:45:03] Using vcs information from agent file: ef94e51a8ea516c.xml
[18:45:03] Checkout directory: /home/teamcity/buildAgent/work/ef94e51a8ea516c
[18:45:03] ▶ Updating sources: auto checkout (on agent) (1s)
[18:45:05] ▶ Resolving artifact dependencies
[18:45:05] ▶ Step 1/1: Prepare Reporter Lab (Command Line) (1s)
[18:45:05] [Step 1/1] Starting: /home/teamcity/buildAgent/temp/agentTmp/custom_script1164548428722804817
[18:45:05] [Step 1/1] in directory: /home/teamcity/buildAgent/work/ef94e51a8ea516c
[18:45:05] [Step 1/1] ansible-playbook 2.7.5
[18:45:05] [Step 1/1] config file = /etc/ansible/ansible.cfg
[18:45:05] [Step 1/1] configured module search path = [u'/home/teamcity/.ansible/plugins/modules', u'/usr/share/ansibl
[18:45:05] [Step 1/1] ansible python module location = /usr/lib/python2.7/dist-packages/ansible
[18:45:05] [Step 1/1] executable location = /usr/bin/ansible-playbook
[18:45:05] [Step 1/1] python version = 2.7.15rc1 (default, Nov 12 2018, 14:31:15) [GCC 7.3.0]
```

Проблемы

- Сложные Powershell скрипты
- У всех разные реализации
- Непонятно, установилось или нет
- Где искать логи?
- Много коммуникации (знаниями обладают лишь несколько человек)

Проблемы

- Сложные Powershell скрипты + ~~Декларативные Ansible конфиги~~
- У всех разные реализации ~~Общие~~ конфигурируемые роли
- Непонятно, установилось или нет ~~Можно посмотреть в TeamCity~~
- Где искать логи? ~~Лог Ansible всегда доступен в TeamCity~~
- Много коммуникации (знаниями обладают лишь несколько человек)
~~Все заинтересованные лица знают, где лежат и как работают плейбуки~~

О чём не говорили

- Infrastructure provisioning: AWS, Azure, VMWare, Openstack

О чём не говорили

- Infrastructure provisioning: AWS, Azure, VMWare, Openstack
- Многие фичи Ansible: Handlers, Tags, etc...

О чём не говорили

- Infrastructure provisioning: AWS, Azure, VMWare, Openstack
- Многие фичи Ansible: Handlers, Tags, etc...
- Деплой на продакшн

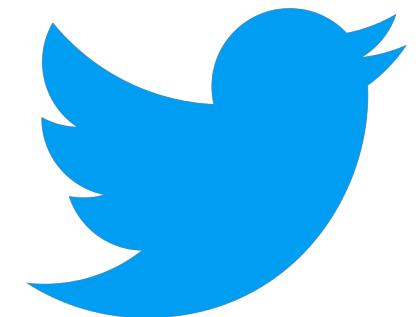
Ссылки

- <https://github.com/kot-lex/spbdotnet-ansible>
- <https://github.com/ansible/ansible/blob/devel/examples/scripts/ConfigureRemotingForAnsible.ps1>
- <https://www.youtube.com/watch?v=FEdXUv02Dbg>
- <https://galaxy.ansible.com/>
- <https://chocolatey.org/>
- https://docs.ansible.com/ansible/latest/dev_guide/developing_modules_general_windows.html

Вопросы?



@kot_lex



@kot_lex



kot.lex@gmail.com