



FOREMAN

FOREMAN/KATELLO MIT ANSIBLE AUTOMATISIEREN

\$ WHOAMI

Evgeni Golov

Senior Software Engineer at Red Hat

ex-Consultant at Red Hat

Debian and Grml Developer

♥ FOSS ♥

♥ automation ♥

AGENDA

- Motivation / WTF
- Warum nicht X?!
- Foreman Ansible Modules
- Workflow Beispiele
- Selber Module schreiben!

MOTIVATION / WTF

WAS IST FOREMAN?

- Tool zur Verwaltung von physikalischen und virtuellen Servern
- Power Management, Installation, Konfiguration
- Bare-Metal, VMware, RHV, OpenStack, GCE, Azure, etc
- Erweiterbar durch Plugins (zB Katello, Monitoring, Ansible)

WAS IST KATELLO?

- Plugin für Foreman
- Content Management (RPM, DEB, Puppet, Containers, Files)
- Content kann gruppiert und gefiltert an Clients ausgeliefert werden
- Erlaubt Snapshots von Content zur Versionierung
- Patch Management

WAS IST ANSIBLE?

- "radically simple IT automation engine"
- bringt eine enorme Zahl an Modulen für unterschiedliche Einsatzzwecke mit
- kann leicht durch eigene Module erweitert werden
- lässt sich gut mit REST APIs integrieren

WARUM AUTOMATISIEREN?

- Jeder kann mitmachen
- Peer Review der Änderungen
- Rollback bei Problemen
- Reproducibility
- Daten lesbar speichern und versionieren

WIE AUTOMATISIEREN?

FOREMAN HAT EINE WEBUI!

- Kein Review der Änderungen möglich
- Zurückspringen zu älteren Einstellungen aufwändig
- Reproducibility ist eher nicht gegeben


WIE AUTOMATISIEREN?

FOREMAN HAT EINE CLI!

- Ich hab da mal schnell was mit sed und awk gebaut. Nein!
- Kann auch CSV/JSON Output und jq/jo sind toll, aber immer noch nein!
- Ansible command/shell Module 🙄

WIE AUTOMATISIEREN?

FOREMAN HAT EINE API!

- 
- Daten (Zustand) in einer Datenstruktur (YAML/JSON)
- Ein API Client übernimmt die Arbeit
- API Client selber schreiben? Später!
- Ansible!

WARUM NICHT X?!

FOREMAN UND KATELLO MODULE

- Ansible Upstream seit 2.3 (2016)
- Deprecated seit 2.8
- Wird in 2.12 entfernt
- Ein Modul für alles, dadurch kompliziert zu bedienen
- Benutzt die `naïlgun` Bibliothek

```
- name: Create Organization
  foreman:
    username: admin
    password: admin
    server_url: https://foreman.example.com
    entity: organization
    params:
      name: My Cool New Organization
```

```
- name: Enable RHEL Product
  katello:
    username: admin
    password: admin
    server_url: https://katello.example.com
    entity: repository_set
    params:
      name: Red Hat Enterprise Linux 7 Server (RPMs)
      product: Red Hat Enterprise Linux Server
      organization: Default Organization
      basearch: x86_64
      releasever: 7Server
```

ANSIBLE - MODULE - FOREMAN

- Seit 2015 gibt es auch `ansible-module-foreman` von Thomas Krahn (@Nosmoht)
- Gut gepflegt
- Benutzt eine eigene Bibliothek um mit der API zu sprechen
 - Benutzt nicht Foremans `apidoc.json`
 - Braucht Anpassungen für Plugins
- Aktuell kein Katello Support

FOREMAN ANSIBLE MODULES

FOREMAN ANSIBLE MODULES

- Seit Juni 2017
- Teil der Foreman Organisation (Git, Dokumentation)
- Versucht foreman/katello aufzuräumen
- Zunächst durch Aufsplittung in einzelne Module pro Objekt
- Dann durch Aufbau eines Frameworks um Module schlank zu halten
- Tests!

FOREMAN ANSIBLE MODULES

- Die Nutzung von `nailgun` wurde irgendwann anstrengend
 - Welche Version von `nailgun`?
 - Was ist mit Plugins?
 - Funktioniert nicht ohne Katello
- `apypie` Bibliothek als Ersatz für `nailgun`
 - Liest die `apidoc.json` von Foreman
- Durch existierendes Framework und Tests ist die Migration einfach

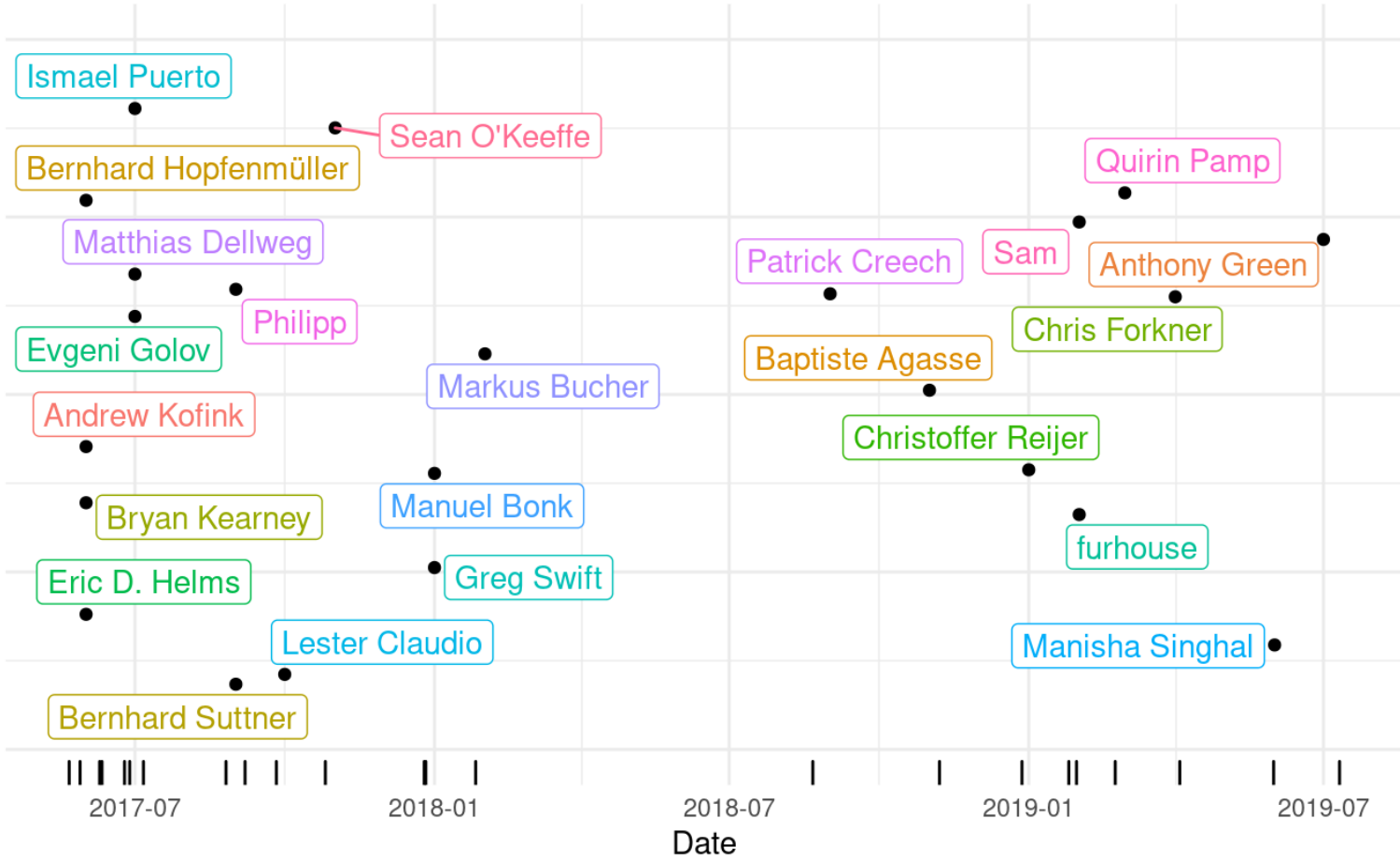
```
- name: "create example.org domain"
  foreman_domain:
    name: "example.org"
    description: "Example Domain"
    server_url: "https://foreman.example.com"
    username: "admin"
    password: "secret"
    state: present
```

```
- name: "Enable RHEL 7 RPMs repositories"
  katello_repository_set:
    username: "admin"
    password: "changeme"
    server_url: "https://foreman.example.com"
    name: "Red Hat Enterprise Linux 7 Server (RPMs)"
    organization: "Default Organization"
    product: "Red Hat Enterprise Linux Server"
    repositories:
      - releasever: "7Server"
        basearch: "x86_64"
    state: enabled
```

FOREMAN ANSIBLE MODULES - STATS

- 50  auf GitHub
- 25 Contributors (10 Red Hat, 7 ATIX)
- 9 neue Contributors in 2019

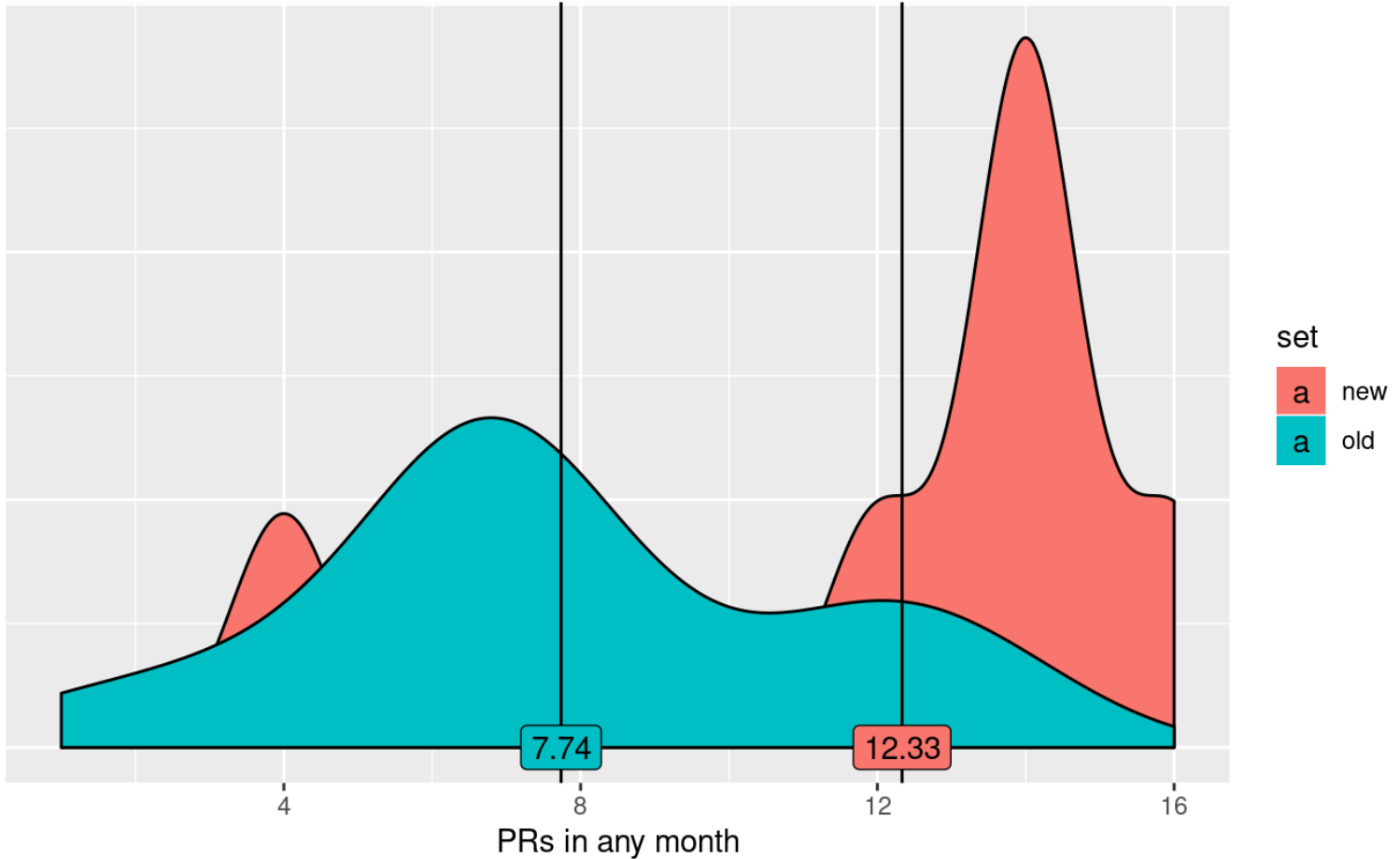
Brand New Contributors per month in foreman-ansible-modules



source: Git repo data

Distribution of monthly PR activity

PR data split into old/new at 2019-02-01



Significant change in PR activity, p-value 0.047

FOREMAN ANSIBLE MODULES - OUTLOOK

- Bald auf Ansible Galaxy
- Bald als RPM auf yum.theforeman.org
- Beide Wege werden Ansible 2.8 benötigen (Ansible Collections)
- Module weiterhin Ansible 2.3+ kompatibel

WORKFLOW BEISPIELE

KATELLO

REPOSITORY+CONTENTVIEW

```
- name: "Enable RHEL repositories"
  katello_repository_set:
    name: "Red Hat Enterprise Linux 7 Server (RPMs)"
    product: "Red Hat Enterprise Linux Server"
    repositories:
      - releasever: "7Server"
        basearch: "x86_64"
        state: enabled
- name: "Create RHEL ContentView"
  katello_content_view:
    name: "RHEL"
    repositories:
      - name: "Red Hat Enterprise Linux 7 Server (RPMs)"
        product: "Red Hat Enterprise Linux Server"
```

KATELLO

SYNC+PUBLISH+PROMOTE

```
- name: "Sync RHEL repositories"
  katello_sync:
    product: "Red Hat Enterprise Linux Server"

- name: "Publish RHEL content view"
  katello_content_view_version:
    content_view: "RHEL"

- name: "Promote RHEL content view to Test"
  katello_content_view_version:
    content_view: "RHEL"
    current_lifecycle_environment: "Library"
  lifecycle_environments:
    - Test
```

KATELLO

SYNC+PUBLISH+PROMOTE

```
- name: "Sync RHEL repositories"
  katello_sync:
    product: "Red Hat Enterprise Linux Server"

- name: "Publish and promote RHEL content view"
  katello_content_view_version:
    content_view: "RHEL"
  lifecycle_environments:
    - Library
    - Test
```

KATELLO LIFECYCLE ENVIRONMENT + ACTIVATION KEY

```
- katello_lifecycle_environment:  
  name: "{{ lifecycle_env }}"  
  prior: "Library"  
  
- name: "Copy Activation Key"  
  katello_activation_key:  
    name: "{{ activation_key }}"  
    new_name: "{{ activation_key }}-{{ lifecycle_env }}"  
    state: 'copied'  
  
- name: "Set Lifecycle Environment for Activation Key"  
  katello_activation_key:  
    name: "{{ activation_key }}-{{ lifecycle_env }}"  
    lifecycle_environment: "{{ lifecycle_env }}"
```

FOREMAN CLEANUP

```
- name: "Clean all media"
  foreman_installation_medium:
    name: "*"
    state: absent
- name: "Dissociate all Provisioning templates"
  foreman_provisioning_template:
    name: "*"
    organizations: []
    locations: []
- name: "Dissociate all Partition Table templates"
  foreman_ptable:
    name: "*"
    organizations: []
    locations: []
```

**SELBER MODULE
SCHREIBEN!**

MODUL AUFBAU

Die meisten Module sind dazu da Objekte in Foreman zu verwalten

1. Bereits vorhandenes Objekt suchen
2. Objekt mit den vom User gegebenen Daten vergleichen
3. Objekt speichern

Dafür gibt es ein Framework...

Wir haben einen Wrapper um AnsibleModule:

```
from ansible.module_utils.foreman_helper import
    ForemanEntityApyieAnsibleModule

module = ForemanEntityApyieAnsibleModule(
    entity_spec=dict(name=dict(required=True)))
```

Parameter laden und API Verbindung testen:

```
entity_dict = module.clean_params()  
module.connect()
```

Bereits existierendes Objekt finden und es mit den per Parameter übergebenen Daten updaten:

```
entity = module.find_resource_by_name('architectures',  
    name=entity_dict['name'], failsafe=True)  
changed = module.ensure_entity('architectures',  
    entity_dict, entity)  
module.exit_json(changed=changed)
```

```
from ansible.module_utils.foreman_helper import
    ForemanEntityApyPieAnsibleModule

module = ForemanEntityApyPieAnsibleModule(
    entity_spec=dict(name=dict(required=True)))
entity_dict = module.clean_params()
module.connect()

entity = module.find_resource_by_name('architectures',
    name=entity_dict['name'], failsafe=True)
changed = module.ensure_entity('architectures',
    entity_dict, entity)
module.exit_json(changed=changed)
```

Nur der Name der Architektur ist ja langweilig...

```
module = ForemanEntityApyPieAnsibleModule(  
    entity_spec=dict(  
        name=dict(required=True),  
        operatingsystems=dict(type='entity_list',  
                               flat_name='operatingsystem_ids'),  
    ),  
)
```

```
if not module.desired_absent:
    if 'operatingsystems' in entity_dict:
        entity_dict['operatingsystems'] =
            module.find_operatingsystems(
                entity_dict['operatingsystems'], thin=True)
```

```
if not module.desired_absent:
    if 'operatingsystems' in entity_dict:
        search_list = ["title~{}".format(title) for title
                        in entity_dict['operatingsystems']]
    entity_dict['operatingsystems'] =
        module.find_resources('operatingsystems', search_list,
                              thin=True)
```


LINKS

docs: theforeman.org/plugins/foreman-ansible-modules

git: github.com/theforeman/foreman-ansible-modules

migration: [issue #274](#)

THANKS!

 evgeni@golov.de

 die-welt.net

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

 [@zhenech@chaos.social](https://discord.com/users/zhenech@chaos.social)

 [@evgeni](https://github.com/evgeni)

 [zhenech](https://t.me/zhenech)