

IT Automation with Ansible



Dr. Bernhard Hopfenmüller

19.10.2017

How complicated is complicated enough?



<http://www.hldataprotection.com/2010/09/articles/consumer-privacy/if-the-online-notice-is-too-complex-does-that-open-the-door-to-tort-claims/>

How good is IT Automation...



<https://depositphotos.com/16900495/stock-photo-good-luck.html>

...if it is too complicated?

Ansible ...

- ▶ ... is „radically simple“
- ▶ ... requires: 1 Unixnode, SSH, Python
- ▶ ... needs no agent
- ▶ ... describes the desired state
- ▶ ... is written in YAML
- ▶ ... satisfies idempotency



- ▶ User: Write Statements
- ▶ Ansible: Create program
- ▶ Ansible: Copy program to Run-Node
- ▶ Ansible: Connect to Run-Node (local, SSH, PowerShell)
- ▶ Ansible: Run Program
- ▶ Ansible: Communication via JSON
- ▶ Ansible: Delete local Program
- ▶ Ansible: Give Feedback to User



- ▶ **Inventory** - infrastructure as code
- ▶ **Adhoc mode** - one task + terminal
- ▶ **Playbook** - collection of tasks/roles
- ▶ **Role** - one collection of tasks
- ▶ **Task** - one Ansible command:
'apt-get install nano'
- ▶ **Jinja 2 templating** - e.g. for variables
- ▶ **Modules** - core-components:
yum module, ping module,...
communication via JSON

```
[dbserver]
db1.server.com
db2.server.com

[webserver]
web1.server.com ansible_host=10.11.12.13
web2.server.com ansible_host=10.11.12.14

[appserver]
app1.server.com default_port=9418
app2.server.com default_port=42

[linuxserver:children]
dbserver
webserver
appserver
```

- ▶ defining the infrastructure
- ▶ ini or yaml syntax
- ▶ combine servers into groups
- ▶ define specific vars

- ▶ run only one task quickly
- ▶ either one-timer or quicker than playbook

```
$ ansible webserver -m patch \
-a \
"src=/src/critical.patch \
dest=/app/bin/base"

$ ansible webserver -a "sbin/reboot"

$ ansible webserver -m ping
```



```
--  
- hosts: webserver # where shall I run?  
  become: true     # run as root  
  roles:  
    - bake_webserver # role 1  
    - install_gitlab # role 2
```

- ▶ combination of roles
- ▶ written in YAML (or JSON)-Syntax
- ▶ ideally in combination with VCS

Setup a gitlab server

- ▶ combination of tasks
- ▶ featuring idempotency
- ▶ ideally in combination with VCS

```
- name: Install GitLab dependencies.
  yum:
    name: {{ item }}
    state: installed
  with_items:
    - openssh-server
    - postfix
    - curl
    - openssl

- name: Install GitLab
  yum:
    name: {{ gitlab_edition }}
    state: present

- name: Copy GitLab configuration file.
  template:
    src: gitlab.rb.j2
    dest: /etc/gitlab/gitlab.rb
    owner: root
    group: root
    mode: 0600
  notify: restart gitlab
```

```
- name: Install GitLab dependencies.
  yum:
    name: {{ item }}
    state: installed
  with_items:
    - openssh-server
    - postfix
    - curl
    - openssl

- name: Install GitLab
  yum:
    name: {{ gitlab_edition }}
    state: present

- name: Copy GitLab configuration file.
  template:
    src: gitlab.rb.j2
    dest: /etc/gitlab/gitlab.rb
    owner: root
    group: root
    mode: 0600
  notify: restart gitlab
```

ATIX and Ansible - a truly great combination



Engineering for and with Ansible



+

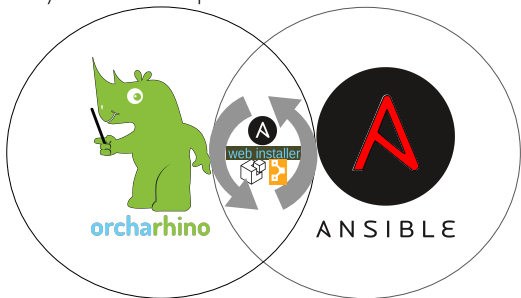


FOREMAN

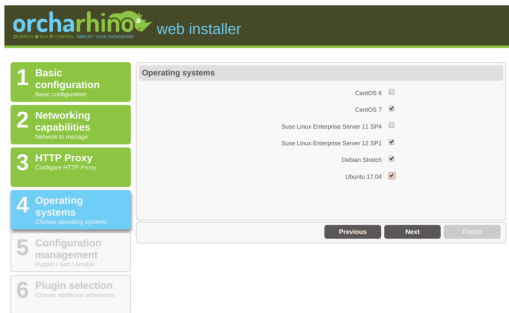
- ▶ Active Member of the OpenSource Community
- ▶ Ansible Modules for Foreman (amongst main contributors)
- ▶ More modules under development
- ▶ Ansible as daily QA for orcharhino

Configuration Management for lazy smart DevOps

- ▶ convenient and easy setup
- ▶ setup 100 % reproducible
- ▶ any combination is possible
- ▶ easily extendable



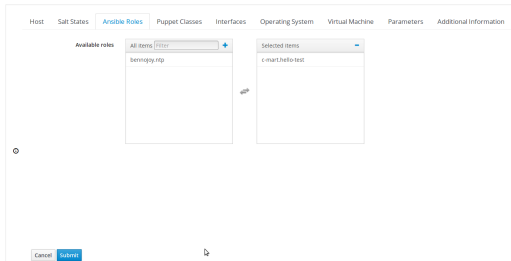
Webinstaller



The screenshot shows the 'orcharhino web installer' interface. On the left is a vertical sidebar with six steps: 1. Basic configuration (Basic configuration), 2. Networking capabilities (Network to manage), 3. HTTP Proxy (Configure HTTP Proxy), 4. Operating systems (Choose operating systems), 5. Configuration management (Puppet / Salt / Ansible), and 6. Plugin selection (Choose additional extensions). Step 4 is currently selected. The main content area is titled 'Operating systems' and lists five options with checkboxes: CentOS 6, CentOS 7 (checked), Suse Linux Enterprise Server 11 SP4, Suse Linux Enterprise Server 12 SP1 (checked), Debian Stretch (checked), and Ubuntu 17.04 (checked). At the bottom of the main area are three buttons: 'Previous', 'Next', and 'Finish'.

- ▶ Installer for orcharhino
- ▶ Configure according to your needs
- ▶ Installer writes config
- ▶ Runs playbook
- ▶ Idempotency ↔ Customize and rerun

Use Ansible within orcharhino



- ▶ Run playbooks from orcharhino
- ▶ in combination with Puppet or standalone
- ▶ under active developement

Host Salt States **Ansible Roles** Puppet Classes Interfaces Operating System Virtual Machine Parameters Additional Information

Available roles

All Items +

bennojoy.ntp

→

Selected Items -

c-mart.hello-test

⊙

Cancel Submit

Task

Running Steps

Errors

Locks

Raw

Stop auto-reloading

Dynflow console

Resume

Cancel

Name:

Play Ansible roles on host judy-bickmore.stage....

Result:

Ⓢ

Triggered by:

admin

Execution type:

Immediate

Start at:

less than a minute ago

Started at:

less than a minute ago

Ended at:

N/A

Start before:

-

State: running

0% Complete

Output:

PLAY [all] *****

TASK [Gathering Facts] *****

Task

Running Steps

Errors

Locks

Raw

Stop auto-reloading

Dynflow console

Resume

Cancel

Name:

Play Ansible roles on host judy-bickmore.stage....

Result:

Ⓢ

Triggered by:

admin

Execution type:

Immediate

Start at:

less than a minute ago

Started at:

less than a minute ago

Ended at:

N/A

Start before:

-

State: running

25% Complete

Output:

PLAY [all] *****

TASK [Gathering Facts] *****

Task Running Steps Errors Locks Raw

Start auto-reloading Dynflow console Resume Cancel

Name: Play Ansible roles on host judy-bickmore.stage....

Result: success

Triggered by: admin

Execution type: immediate

Start at: less than a minute ago

Started at: less than a minute ago

Ended at: less than a minute ago

Start before: -

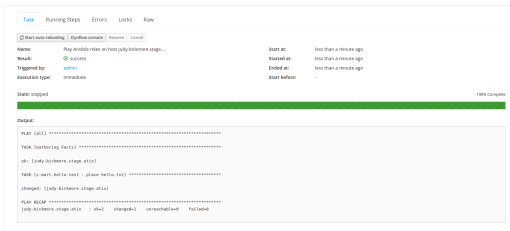
State: stopped

100% Complete

Output:

```
PLAY [all] *****
TASK [Gathering Facts] *****
ok: [judy-bickmore.stage.atix]
TASK [c-mart.hello-test : place hello.txt] *****
changed: [judy-bickmore.stage.atix]
PLAY RECAP *****
judy-bickmore.stage.atix : ok=2  changed=1  unreachable=0  failed=0
```

Use Ansible within orcharhino



The screenshot shows the 'Task' tab of an Ansible run in orcharhino. The task is named 'Play Ansible roles on host jedy-bickmore.stage...'. It was triggered by 'admin' and executed immediately. The status is 'Success'. The output shows that the 'jedy-bickmore.stage.atix' role was applied successfully, changing the 'hello' variable to 'atix'.

```
PLAY [all] .....
TASK [Gathering Facts] .....
ok: [jedy-bickmore.stage.atix]
TASK [include_role: jedy-bickmore.stage.atix] .....
changed: [jedy-bickmore.stage.atix]
PLAY RECAP .....
jedy-bickmore.stage.atix : ok=2 changed=1 unreachable=0 failed=0
```

- ▶ Run playbooks from orcharhino
- ▶ in combination with Puppet or standalone
- ▶ under active development by community

Example projects



- ▶ (one time) Deployment of infrastructures
- ▶ interface to virtualization
- ▶ CI - CD integration
- ▶ VCS integration
- ▶ Springboot framework
- ▶ Compliance Management
- ▶ ...

- ▶ From end of 2017, Begin of 2018
- ▶ 3 days
- ▶ Hands-on based
- ▶ topics (can) include
 - ▶ Ansible basics
 - ▶ From Paper to Playbook
 - ▶ Jinja Templating
 - ▶ Ansible + VCS
 - ▶ (Tower)



Ansible - Simple and versatile IT Automation



- ▶ Engineering
- ▶ Consulting
- ▶ Training
- ▶ Support