IT Automation with Ansible

Dr. Bernhard Hopfenmüller

19.10.2017
How complicated is complicated enough?

How good is IT Automation...

...if it is too complicated?

https://depositphotos.com/16900495/stock-photo-good-luck.html
Ansible in a nutshell

Ansible ...

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

- 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
Ansible dict

- **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
Inventory

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

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

[webservice]
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
webservice
appserver
```
Ad-hoc mode

- 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
```
Playbook

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

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

Setup a gitlab server

```yaml
- 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
```
Role

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

- name: Install GitLab
  yum:
    name: {{ gitlabEdition }}
    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
Ansible meets Engineering

Engineering for and with Ansible

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

#atix #osad2017
orcharhino and Ansible

Configuration Management for *lazy* smart DevOps

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

#atix #osad2017
orcharhino and Ansible

Webinstaller

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

#atix #osad2017
orcharhino and Ansible

Use Ansible within orcharhino

- Run playbooks from orcharhino
- in combination with Puppet or standalone
- under active development
# orcharchino and Ansible

<table>
<thead>
<tr>
<th>Available roles</th>
<th>Selected items</th>
</tr>
</thead>
<tbody>
<tr>
<td>c-mart.hello-test</td>
<td></td>
</tr>
<tr>
<td>bennioy.ntp</td>
<td></td>
</tr>
</tbody>
</table>
orchartino and Ansible

<table>
<thead>
<tr>
<th>Task</th>
<th>Running Steps</th>
<th>Errors</th>
<th>Locks</th>
<th>Raw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop auto-reloading</td>
<td>Dynflow console</td>
<td>Resume</td>
<td>Cancel</td>
<td></td>
</tr>
</tbody>
</table>

**Name:** Play Ansible roles on host judy-bickmore.stage....  
**Result:** ①  
**Triggered by:** admin  
**Execution type:** Immediate

**State:** running  
**0% Complete**

**Output:**

```
PLAY [all] *****************************************************
TASK [Gathering Facts] ****************************************
```
orcharhino and Ansible

<table>
<thead>
<tr>
<th>Task</th>
<th>Running Steps</th>
<th>Errors</th>
<th>Locks</th>
<th>Raw</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stop auto-reloading</td>
<td>Dynflow console</td>
<td>Resume</td>
<td>Cancel</td>
</tr>
<tr>
<td>Name:</td>
<td>Play Ansible roles on host judy-bickmore.stage....</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Result:</td>
<td>©</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triggered by:</td>
<td>admin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execution type:</td>
<td>Immediate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State:</td>
<td>running</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Start at: less than a minute ago
Started at: less than a minute ago
Ended at: N/A
Start before: -

Output:

PLAY [all] *******************************************************
TASK [Gathering Facts] *******************************************************
orcharhino and Ansible
orcharhino and Ansible

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

#atix #osad2017
Example projects

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

#atix #osad2017
Ansible Training

- 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