

Ansible Module Development



Ansible Module Development

- Module? What?
- Write and test a module?
- Get the module into Ansible?

About me

- René Moser (resmo)
 - Ansible Core Member
 - Author of Ansible CloudStack modules
 - Ansible user since ~v0.7
 - System Engineer @ SWISS TXT

Modules?

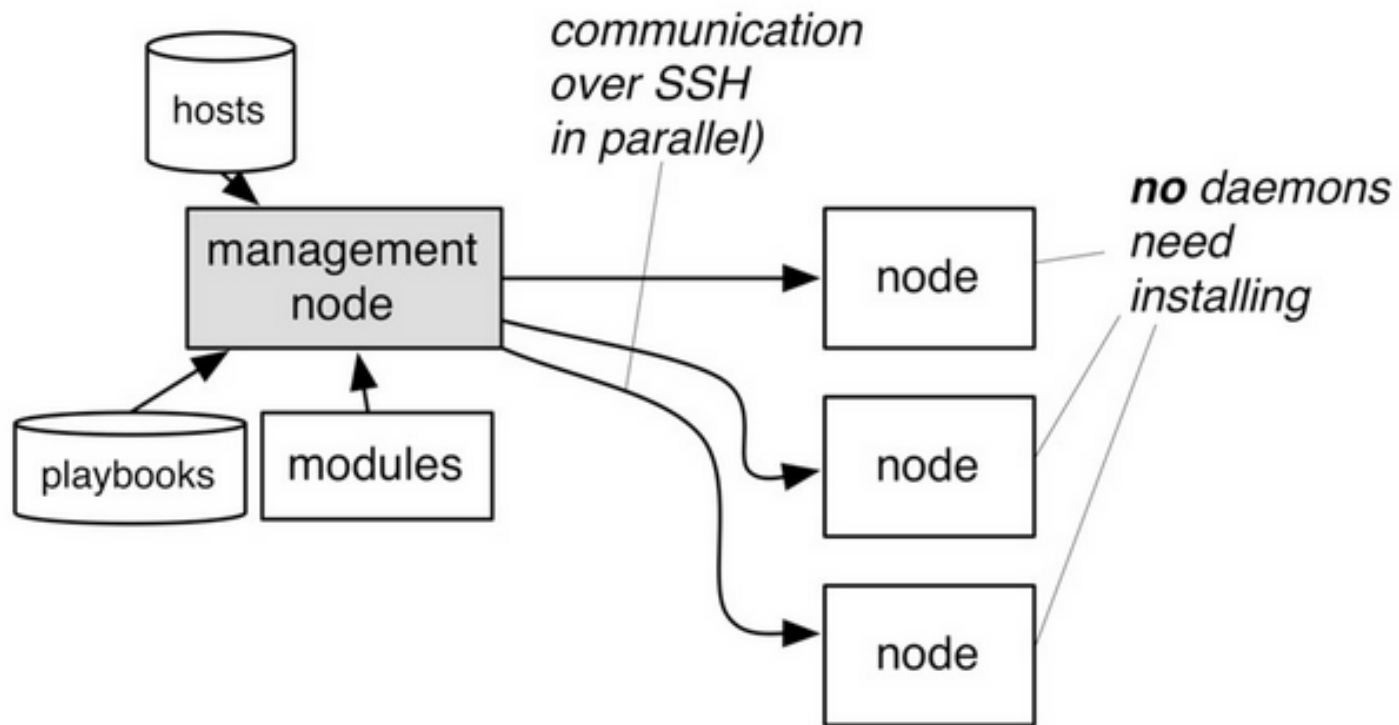
```
1  ---
2  - name: install search engine
3    .. remote_user: root
4    .. hosts: search
5    .. tasks:
6      .. - name: add yacy gpg key
7        .... apt_key: url='http://debian.yacy.net/yacy_orbiter_key.asc'
8
9      .. - name: add yacy repo
10        .... apt_repository: repo='deb http://debian.yacy.net/'
11
12      .. - name: install packages
13        .... apt: pkg={{ .item }}
14        .... with_items:
15          ..... - openjdk-7-jre-headless
16          ..... - yacy
17          ..... - nginx
18
19      .. - name: configure nginx for yacy
20        .... template: src=files/nginx/default.j2 dest=/etc/nginx/sites-available/default
21        .... notify: restart nginx
22
23      .. - name: ensure services are running
24        .... service: name={{ .item }} state=started enabled=yes
25        .... with_items:
26          ..... - nginx
27          ..... - yacy
28
```

Modules

- **Addon** to Ansible
- Runs on **target**
- Can have **arguments**
- Returns result e.g. **state**

How modules work

Modus operandi



Why Modules?

- Solve a **complex** problem
- Ensure a **state**
- **User** interface

Bad Modules

- Could be **replaced by command** module
- Code is **hard to read** and not tested
- **Not** handle **idempotence**

Modules written in?

- **Any** language (python, perl, bash, powershell)
- Binary modules in 2.2 (golang, c)

Modules in Python

- Python version compatible
 - 2.4 if possible
 - 2.6 =< are accepted
 - 3 =< may be accepted soon
- Code style **pep8**-ish
 - Line length not enforced
 - Utils imports nearby main() and not on top

Get Started

- Docs:
 - http://docs.ansible.com/ansible/developing_modules.html
- See **how others** did:
 - <https://github.com/ansible/ansible-modules-core>
 - <https://github.com/ansible/ansible-modules-extras>
- Contact humans related to development
 - IRC: <irc://irc.freenode.net/ansible-devel>
 - ML: <https://groups.google.com/forum/#!forum/ansible-devel>

Module Helpers

- Ansible: `/lib/ansible/module_utils/`
 - **basic.py**
 - `urls.py`
 - `api.py`
 - `facts.py`

Args?

```
1 ---
2 - name: install.search.engine
3   remote_user: root
4   hosts: search
5   tasks:
6     - name: add.yacy.gpg.key
7       apt_key: url='http://debian.yacy.net/yacy_orbiter_key.asc'
8
9     - name: add.yacy.repo
10      apt_repository: repo='deb http://debian.yacy.net /'
11
12    - name: install.packages
13      apt: pkg={{ item }}
14      with_items:
15        - openjdk-7-jre-headless
16        - yacy
17        - nginx
18
19    - name: configure.nginx.for.yacy
20      template: src=files/nginx/default.j2 dest=/etc/nginx/sites-available/default
21      notify: restart.nginx
22
23    - name: ensure.services.are.running
24      service: name={{ item }} state=started enabled=yes
25      with_items:
26        - nginx
27        - yacy
28
```

Basic.py

- Argument specification aka “**arg spec**”
 - Required
 - Default (e.g. default='present')
 - Choices (a list of possible values)
 - Type (str, int, bool, list, dict, path)

Basic.py

- Argument **relation** handling:
 - required_together
 - required_if
 - required_one_of
 - mutually_exclusive

Basic.py

- **Json** returns
 - fail_json()
 - exit_json()
- **Never** ever **use**:
 - Print!
 - sys.exit()

notification/irc.py

```
246 def main():
247     ... module = AnsibleModule(
248         ... argument_spec=dict(
249             ... server=dict(default='localhost'),
250             ... port=dict(default=6667),
251             ... nick=dict(default='ansible'),
252             ... nick_to=dict(required=False, type='list'),
253             ... msg=dict(required=True),
254             ... color=dict(default="none", aliases=['colour'], choices=["white", "black", "blue",
255             ... "green", "red", "brown",
256             ... "purple", "orange", "yellow",
257             ... "light_green", "teal", "light_cyan",
258             ... "light_blue", "pink", "gray",
259             ... "light_gray", "none"]),
260             ... style=dict(default="none", choices=["underline", "reverse", "bold", "italic",
261             ... "none"]),
262             ... channel=dict(required=False),
263             ... key=dict(no_log=True),
264             ... topic=dict(),
265             ... passwd=dict(no_log=True),
266             ... timeout=dict(type='int', default=30),
267             ... part=dict(type='bool', default=True),
268             ... use_ssl=dict(type='bool', default=False)
269             ... ),
270             ... supports_check_mode=True,
271             ... required_one_of=[['channel', 'nick_to']]
272         ... )
```

Test a module

- Running **Ansible** from a **git checkout**
 - `./hacking/env-setup`
- Run a module **locally**
 - `./hacking/test-module -m <module> -a <args>`

Demo

Custom module integration

- env **ANSIBLE_LIBRARY** = <path>
- **./library** alongside your playbooks
- **library** = <path> in **ansible.cfg**
 - http://docs.ansible.com/ansible/intro_configuration.html#library

Integration Tests

- test/integration/
- make cloudstack TEST_FLAGS="--tags test_cs_domain"
- Assert module to **verify to returns**

One test case

```
10 - name: test.create.instance
11   cs_instance:
12     name: "{{.cs_resource_prefix}}-vm-{{.instance_number}}"
13     template: "{{.test_cs_instance_template}}"
14     service_offering: "{{.test_cs_instance_offering_1}}"
15     affinity_group: "{{.cs_resource_prefix}}-ag"
16     security_group: "{{.cs_resource_prefix}}-sg"
17     ssh_key: "{{.cs_resource_prefix}}-sshkey"
18     tags: []
19   register: instance
20 - name: verify.create.instance
21   assert:
22     that:
23     - instance|success
24     - instance|changed
25     - instance.name == "{{.cs_resource_prefix}}-vm-{{.instance_number}}"
26     - instance.display_name == "{{.cs_resource_prefix}}-vm-{{.instance_number}}"
27     - instance.service_offering == "{{.test_cs_instance_offering_1}}"
28     - instance.state == "Running"
29     - instance.ssh_key == "{{.cs_resource_prefix}}-sshkey"
30     - not instance.tags
```

Contribute to Upstream

- → ansible-module-extras repo
- Checklist
 - http://docs.ansible.com/ansible/developing_modules.html#module-checklist
- Travis build
- Github bot “Ansibullbot”

New module PRs

- Need **2 reviewers** / testers policy
- Become a **maintainers** of existing module
 - Verify Changes
 - Help fixing issues
 - Be responsive
- We **need you!** All of you!

Questions?

- Twitter [@resmo79](#)
- GitHub: github.com/resmo
- Blog: www.renemoser.net
- GPG: 0x8306965BE6D6331D