



Une nouvelle API pour les résultats de vos playbooks Ansible

Avril 2019

David Moreau-Simard

```
$ whoami
```

```
David Moreau-Simard
```

```
Senior Software Engineer @ Red Hat
```

- dmsimard sur Twitter, GitHub et freenode
- Dev/Ops et CI/CD: OpenStack, RDO, Ansible et Zuul
- <3 oiseaux

\$ whoami

David Moreau-Simard

Senior Software Engineer @ Red Hat

- dmsimard sur Twitter, GitHub et freenode
- Dev/Ops et CI/CD: OpenStack, RDO, Ansible et Zuul
- <3 oiseaux





ANSIBLE

Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy.

Avoid writing scripts or custom code to deploy and update your applications – automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

<https://github.com/ansible/ansible>

<https://ara.recordsansible.org>

```
$ ansible --help
```

```
./hosts
```

```
# Simple static Ansible inventory  
webserver ansible_host=172.29.239.10 ansible_user=ansible
```

```
# Update all packages on webserver  
$ ansible -i hosts -m package -a "name=* state=latest" webserver  
webserver | SUCCESS => {  
  "ansible_facts": {  
    "pkg_mgr": "dnf"  
  },  
  "changed": false,  
  "msg": "Nothing to do",  
  "rc": 0,  
  "results": []  
}
```

```
$ ansible-playbook --help
```

```
update-packages.yaml
```

```
- name: Update packages on all servers
  hosts: all
  gather_facts: yes
  tasks:
    - name: Update packages
      become: yes
      package:
        name: "*"
        state: latest
      register: updates

    - name: Send an email if there are updates
      mail:
        to: ops@example.org
        subject: "Package updates: {{ ansible_hostname }}"
        body: "{{ updates.stdout }}"
      when: updates is changed
```

\$ ansible-playbook --help

update-packages.yaml

```
- name: Update packages on all servers
  hosts: all
  gather_facts: yes
  tasks:
    - name: Update packages
      become: yes
      package:
        name: "*"
        state: latest
      register: updates

    - name: Send an email if there are updates
      mail:
        to: ops@example.org
        subject: "Package updates: {{ ansible_
        body: "{{ updates.stdout }}"
      when: updates is changed
```

```
Welcome to fish, the friendly interactive shell
[dmsimard@localhost ~/dev/sandbox]$
. ~/dev/virtualenvs/ansible/bin/activate.fish |
```



TL;DR

ARA Records Ansible records Ansible playbooks.
It makes them easier to understand and troubleshoot.



Quickstart (v0.x)

The screenshot shows the ARA web interface with a modal window displaying the following task details:

- Task: **Install latest kernel**
- Host: **centos**
- Time: **2018-11-23 22:29:47**
- Ansible version: **2.7.1**
- Result #1

```
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* base: ewr.edge.kernel.org
...
Package kernel-3.10.0-862.14.4.el7.x86_64
kernel-3.10.0-862.14.4.el7.x86_64
...
cut-config-generic-033-240.el7.x86_64
cut-network-033-240.el7.x86_64
cut-config-rescue-033-240.el7.x86_64
...
3-535.el7_5.1.x86_64
7_5 will be installed
...
updated
...
be an update
updated
...
be an update
```

```
# Installer ARA depuis PyPi
$ pip install ara

# Configurer Ansible pour utiliser ARA
$ export ANSIBLE_CALLBACK_PLUGINS=$(python -m ara.setup.callback_plugins)

# Exécutez votre playbook
$ ansible-playbook playbook.yml

# Démarrez le serveur web de développement
$ ara-manage runserver

# Naviguez à http://127.0.0.1:9191
```



Quickstart (v1.0)

```
GET /api/v1/results/3
```

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
{
  "id": 3,
  "playbook": {
    "id": 1,
    "duration": "19.713072",
    "started": "2019-03-20T19:45:51",
    "ended": "2019-03-20T19:46:11.5",
    "name": "Smoke tests",
    "status": "failed",
```

```
# Installer ARA 1.0 depuis PyPi
$ pip install --pre ara

# Configurer Ansible pour utiliser ARA
$ export ANSIBLE_CALLBACK_PLUGINS=$(python -m ara.setup.callback_plugins)

# Exécutez votre playbook
$ ansible-playbook playbook.yml

# Démarrez le serveur web de développement
$ ara-manage runserver

# Naviguez à http://127.0.0.1:8000/api/v1/
```

Api Root

OPTIONS

GET

The default basic root view for DefaultRouter

```
GET /api/v1/
```

```
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

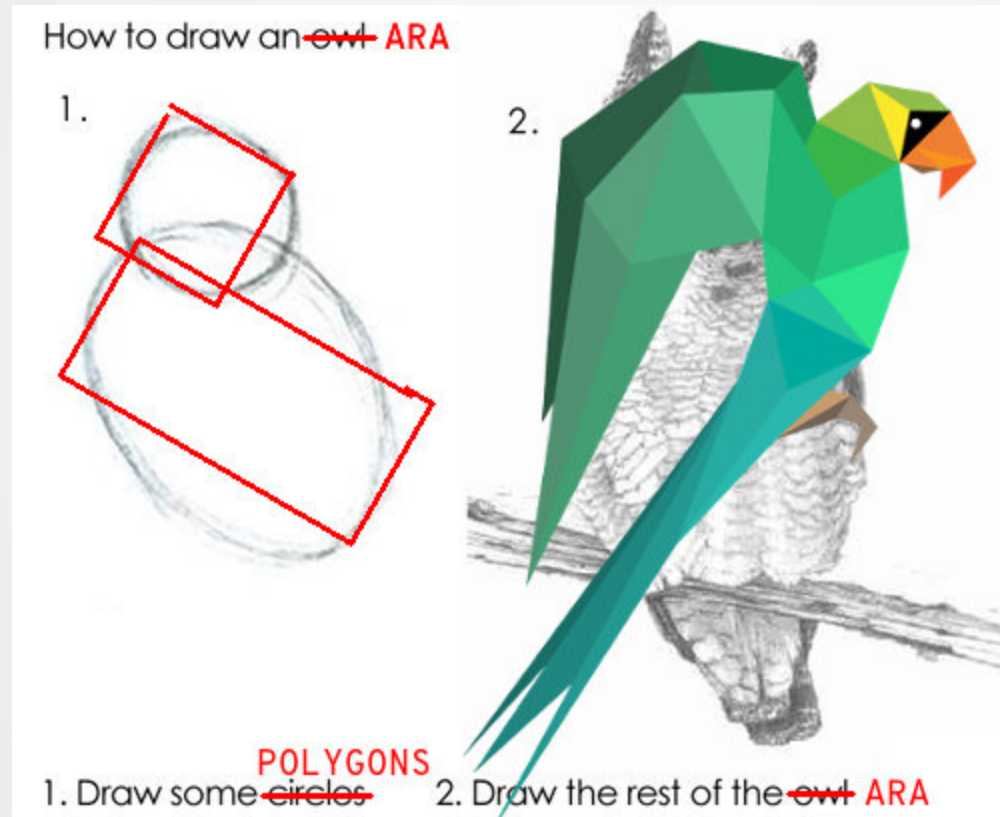
```
{
  "labels": "http://127.0.0.1:8000/api/v1/labels",
  "playbooks": "http://127.0.0.1:8000/api/v1/playbooks",
  "plays": "http://127.0.0.1:8000/api/v1/plays",
  "tasks": "http://127.0.0.1:8000/api/v1/tasks",
  "hosts": "http://127.0.0.1:8000/api/v1/hosts",
  "results": "http://127.0.0.1:8000/api/v1/results",
  "files": "http://127.0.0.1:8000/api/v1/files",
  "records": "http://127.0.0.1:8000/api/v1/records",
  "info": "http://127.0.0.1:8000/api/v1/info"
}
```

La simplicité est une feature

> <https://ara.readthedocs.io/en/latest/manifesto.html>

- 1) *Simplicity is fundamental*
- 2) *Do one thing and do it well*
- 3) *Empower users to get their work done*
- 4) *Don't require users to change their workflows*
- 5) *De-centralized, offline and standalone by default*

Comment ça marche ?



ARA fourni un callback plugin Ansible

```
# https://github.com/ansible/ansible/blob/devel/lib/ansible/plugins/callback
def v2_on_any(self, *args, **kwargs):
def v2_runner_on_failed(self, result, ignore_errors=False):
def v2_runner_on_ok(self, result):
def v2_runner_on_skipped(self, result):
def v2_runner_on_unreachable(self, result):
def v2_playbook_on_start(self, playbook):
def v2_playbook_on_task_start(self, task, is_conditional):
def v2_playbook_on_handler_task_start(self, task):
def v2_playbook_on_play_start(self, play):
def v2_playbook_on_stats(self, stats):
def v2_playbook_on_include(self, included_file):
[...]
```

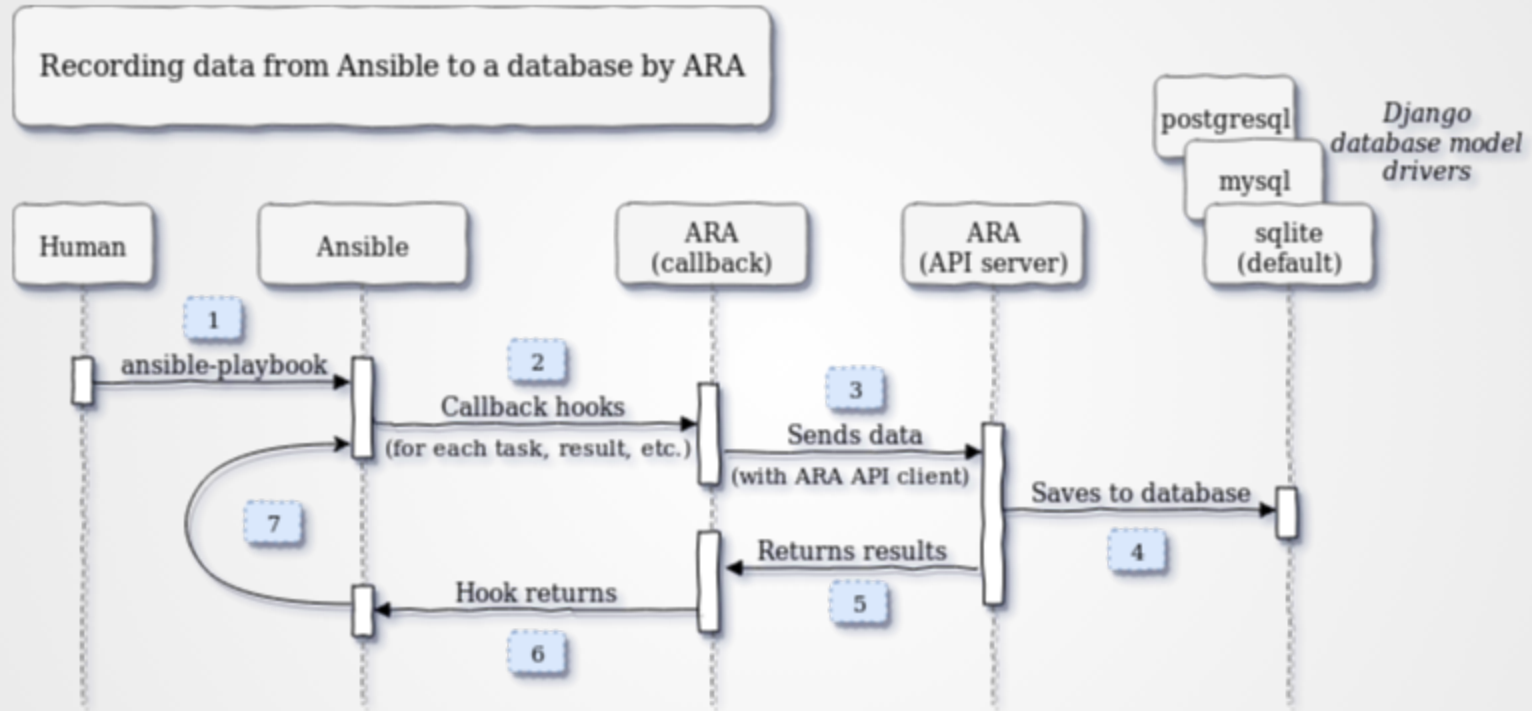
ARA fournì un callback plugin Ansible

```
def v2_playbook_on_start(self, playbook):
    path = os.path.abspath(playbook._file_name)

    # Potentially sanitize some user-specified keys
    for argument in self.ignored_arguments:
        if argument in cli_options:
            cli_options[argument] = "Not saved by ARA as configured by 'ignored_arguments'"

    # Create the playbook
    self.playbook = self.client.post("/api/v1/playbooks",
        ansible_version=ansible_version,
        arguments=cli_options,
        status="running",
        path=path
    )
```

À haut niveau

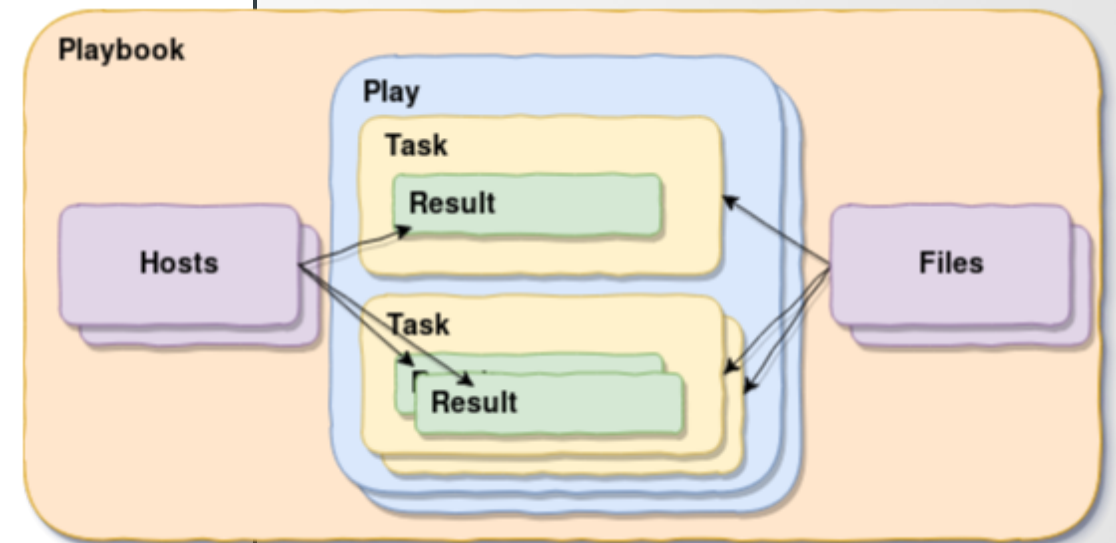


À haut niveau

```
update-packages.yaml
```

```
- name: Update packages on all servers
  hosts: all
  gather_facts: yes
  tasks:
    - name: Update packages
      become: yes
      package:
        name: "*"
        state: latest
      register: updates

- name: Send an email if there are updates
  mail:
    to: ops@example.org
    subject: "Package updates: {{ ansible_hostname }}"
    body: "{{ updates.stdout }}"
  when: updates is changed
```



Utilisation de l'API

example.py

```
#!/usr/bin/env python3
# Different ways to get the ARA API client
import os
from ara.clients.offline import AraOfflineClient
from ara.clients.http import AraHttpClient
from ara.clients import utils as client_utils

offline_client = AraOfflineClient()
http_client = AraHttpClient(endpoint="https://api.demo.recordsansible.org")

api_client = os.environ.get("ARA_API_CLIENT", "offline")
api_server = os.environ.get("ARA_API_SERVER", "http://127.0.0.1:8000")
api_timeout = os.environ.get("ARA_API_TIMEOUT", 30)
client = client_utils.get_client(client=api_client, endpoint=api_server, timeout=api_timeout)
```

Utilisation de l'API

example.py

```
#!/usr/bin/env python3
# Import the client
from ara.clients.http import AraHttpClient

# Instanciate the HTTP client with an endpoint where an API server is listening
client = AraHttpClient(endpoint="https://api.demo.recordsansible.org")

# Get a list of failed playbooks
# /api/v1/playbooks?status=failed
playbooks = client.get("/api/v1/playbooks", status="failed")

# If there are any results from our query, get more information about the
# failure and print something helpful
template = "{timestamp}: {host} failed '{task}' ({task_file}:{lineno})"
for playbook in playbooks["results"]:
    # Get a detailed version of the playbook that provides additional context
    detailed_playbook = client.get("/api/v1/playbooks/%s" % playbook["id"])
```

Utilisation de l'API






```
$ python example.py
2019-03-20T19:46:02.902819: localhost failed 'smoke-tests : Record with no key' (tests/integrat
2019-03-20T19:46:03.780885: localhost failed 'smoke-tests : Record with no value' (tests/integ
2019-03-20T19:46:04.578299: localhost failed 'smoke-tests : Record with invalid type' (tests/
2019-03-20T19:46:06.180167: localhost failed 'smoke-tests : Return false' (tests/integration/
2019-03-20T19:46:33.570741: localhost failed 'fail' (tests/integration/failed.yaml:22)
```

Utilisation de l'API

```
37 + <div className="pf-c-card__body">
38 +   <table class="pf-c-table pf-m-compact pf-m-grid-md" role="grid">
39 +     <thead>
40 +       <tr>
41 +         <th>Name</th>
42 +         <th>OK</th>
43 +         <th>CHANGED</th>
44 +         <th>FAILED</th>
45 +         <th>SKIPPED</th>
46 +         <th>UNREACHABLE</th>
47 +       </tr>
48 +     </thead>
49 +     <tbody>
50 +       {playbook.hosts.map(host => (
51 +         <tr key={host.id}>
52 +           <td data-label="Name">{host.name}</td>
53 +           <td data-label="OK">{host.ok}</td>
54 +           <td data-label="CHANGED">{host.changed}</td>
55 +           <td data-label="FAILED">{host.failed}</td>
56 +           <td data-label="SKIPPED">{host.skipped}</td>
57 +           <td data-label="UNREACHABLE">{host.unreachable}</td>
58 +         </tr>
59 +       )})
60 +     </tbody>
61 +   </table>
62 + </div>
63 + </div>
```

```
68 + {playbook.plays.map(play => (
36 69   <div className="pf-c-card__body">
37 -   <TasksContainer playbook={playbook} />
70 +   <h2>{play.name}</h2>
71 +   <List>
72 +     <ListItem>Started: {play.started}</ListItem>
73 +     <ListItem>Ended: {play.ended}</ListItem>
74 +     <ListItem>Duration: {play.duration}</ListItem>
75 +   </List>
76 +   <table class="pf-c-table pf-m-compact pf-m-grid-md" role="grid">
77 +     <thead>
78 +       <tr>
79 +         <th>Task</th>
80 +         <th>Host</th>
81 +         <th>Action</th>
82 +         <th>Duration</th>
83 +         <th>Status</th>
84 +       </tr>
85 +     </thead>
86 +     <tbody>
87 +       {play.tasks.map(task =>
88 +         task.results.map(result =>
89 +           <tr key={result.id}>
90 +             <th data-label="Task">{task.name}</th>
91 +             <th data-label="Host">{result.host.name}</th>
92 +             <th data-label="Action">{task.action}</th>
93 +             <th data-label="Duration">{result.duration}</th>
94 +             <th data-label="Status">{result.status}</th>
95 +           </tr>
96 +         )})
97 +       </tbody>
98 +     </table>
38 99   </div>
100 + )})
```

Utilisation de l'API

 smoke.yaml	3 Plays	38 Tasks	38 Results	2 Hosts	5 Files	6 Records	 20 sec
 hosts.yaml	2 Plays	6 Tasks	14 Results	5 Hosts	1 Files	0 Records	 6 sec
 import.yaml							

Hosts

Name	OK	CHANGED	FAILED	SKIPPED	UNREACHABLE
localhost	34	9	1	1	0
höstnämē	2	0	0	0	0

Plays

ARA Tasks test play

- Started: 2019-03-20T19:45:52.7
- Ended: 2019-03-20T19:46:08.8
- Duration: 16.713951

Task
smoke-tests : Deferred setup
smoke-tests : include_tasks
smoke-tests : Print normal data

Task	Host	Action	Duration	Status
smoke-tests : Deferred setup	localhost	setup	1.15782	ok
smoke-tests : include_tasks	localhost	include_tasks	0.124497	ok
smoke-tests : Print normal data	localhost	debug	0.116867	ok

API: Use cases

- Reporting
- Troubleshooting
- Monitoring
- Benchmarking / Profiling
- CI / CD
- Intégration avec d'autres systèmes (logstash, grafana, etc.)

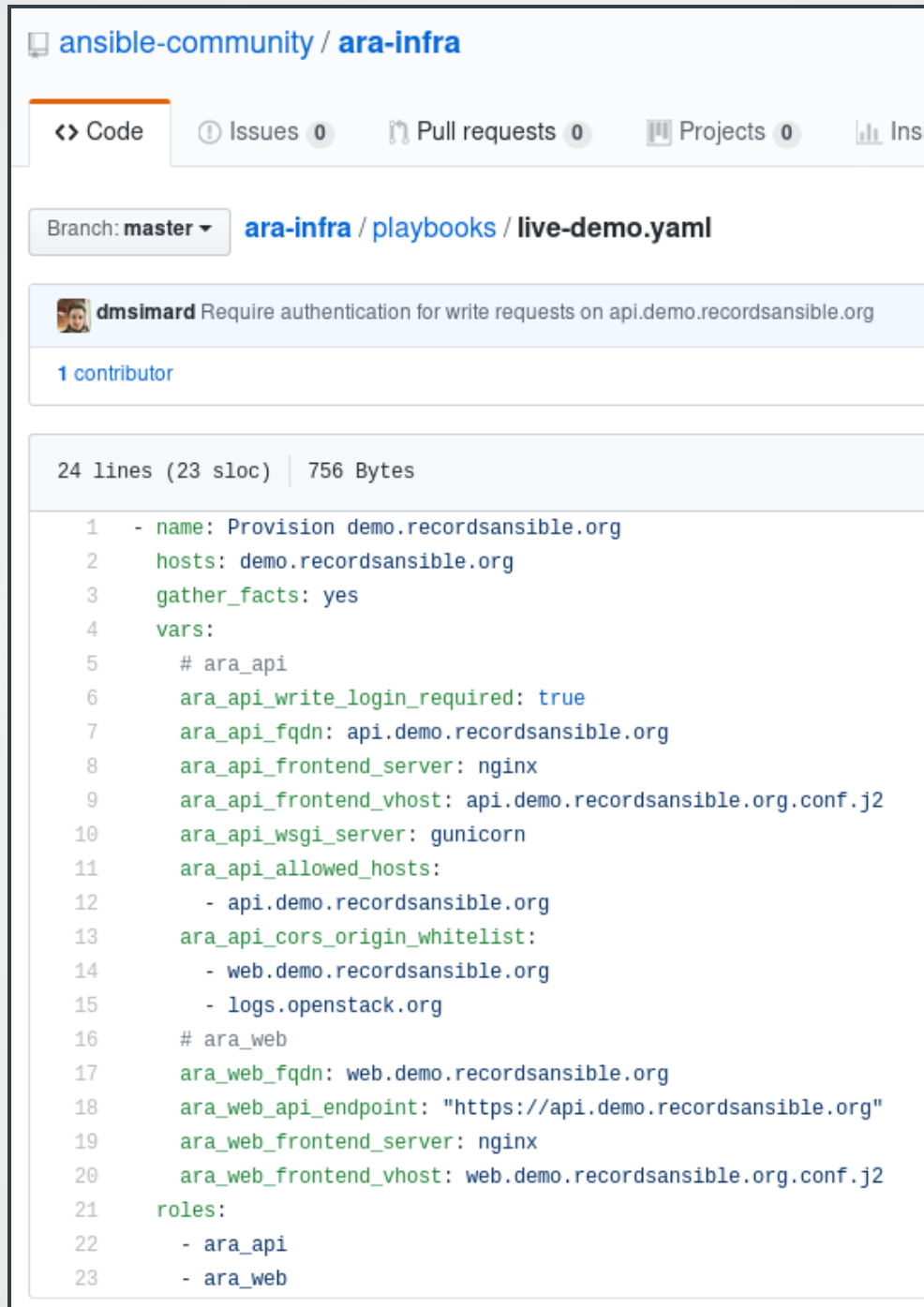


Live demos

<https://api.demo.recordsansible.org/>

<https://web.demo.recordsansible.org/>

Live demos



The screenshot shows a GitHub repository page for `ansible-community / ara-infra`. The page includes navigation tabs for Code, Issues (0), Pull requests (0), Projects (0), and Insights. The current branch is `master`, and the file path is `ara-infra / playbooks / live-demo.yml`. A notification from `dmsimard` states: "Require authentication for write requests on api.demo.recordsansible.org". There is one contributor listed. The file statistics are 24 lines (23 sloc) and 756 Bytes. The file content is a YAML configuration for a play:

```
1 - name: Provision demo.recordsansible.org
2   hosts: demo.recordsansible.org
3   gather_facts: yes
4   vars:
5     # ara_api
6     ara_api_write_login_required: true
7     ara_api_fqdn: api.demo.recordsansible.org
8     ara_api_frontend_server: nginx
9     ara_api_frontend_vhost: api.demo.recordsansible.org.conf.j2
10    ara_api_wsgi_server: gunicorn
11    ara_api_allowed_hosts:
12      - api.demo.recordsansible.org
13    ara_api_cors_origin_whitelist:
14      - web.demo.recordsansible.org
15      - logs.openstack.org
16    # ara_web
17    ara_web_fqdn: web.demo.recordsansible.org
18    ara_web_api_endpoint: "https://api.demo.recordsansible.org"
19    ara_web_frontend_server: nginx
20    ara_web_frontend_vhost: web.demo.recordsansible.org.conf.j2
21  roles:
22    - ara_api
23    - ara_web
```

<https://ara.recordsansible.org>

Avec les roles Ansible inclus

ara_api : Include installation of ARA	fedora-29	include_tasks	0:00:22	0:00:00	OK
ara_api : Prepare git repository for ara	fedora-29	git	0:00:22	0:00:01	CHANGED
ara_api : Install ara	fedora-29	pip	0:00:25	0:00:25	CHANGED
ara_api : Prefix the virtualenv bin directory to PATH	fedora-29	set_fact	0:00:50	0:00:00	OK
ara_api : Include configuration of the ARA API	fedora-29	include_tasks	0:00:51	0:00:00	OK

<https://ara.readthedocs.io/en/feature-1.0/ansible-role-ara-api.html>

ara_web : Install nodejs	ubuntu-bionic	package	0:00:24	0:00:06	CHANGED
ara_web : Include ara-web installation	ubuntu-bionic	include_tasks	0:00:31	0:00:00	OK
ara_web : Ensure libselinux-python is installed for Red Hat derivatives	ubuntu-bionic	package	0:00:31	0:00:00	SKIPPED
ara_web : Ensure git is installed	ubuntu-bionic	package	0:00:31	0:00:01	OK
ara_web : Prepare git repository for ara-web	ubuntu-bionic	git	0:00:33	0:00:01	CHANGED
ara_web : Install ara-web npm dependencies	ubuntu-bionic	npm	0:00:34	0:01:34	CHANGED
ara_web : Configure ara-server API endpoint for ara-web	ubuntu-bionic	copy	0:02:09	0:00:02	CHANGED

<https://ara.readthedocs.io/en/feature-1.0/ansible-role-ara-web.html>

Avec Ansible Tower ou AWX ?



**ANSIBLE
TOWER**
by Red Hat®



En théorie ...



**ANSIBLE
TOWER**
by Red Hat®

- Dashboard
- Jobs
- Schedules
- My View
- RESOURCES
- Templates
- Credentials
- Projects
- Inventories
- Inventory Scripts
- ACCESS
- Organizations

EDIT CONFIGURATION

CONFIGURE TOWER

AUTHENTICATION **JOBS** SYSTEM USER INTERFACE LICENSE

ANSIBLE MODULES ALLOWED FOR AD HOC JOBS REVERT

- command
- shell
- yum
- apt
- apt_key
- apt_repository
- apt_rpm
- service
- group
- user
- mount
- ping
- selinux

PATHS TO EXPOSE TO ISOLATED JOBS REVERT

ANSIBLE CALLBACK PLUGINS REVERT

/var/lib/awx/venv/ara/lib/python2.7/site-packages/ara/plugins/callbacks

DEFAULT JOB TIMEOUT REVERT

0

PER-HOST ANSIBLE FACT CACHE TIMEOUT REVERT

0

EXTRA ENVIRONMENT VARIABLES ?

1 {}



```
# Create a new empty virtualenv that doesn't inherit system packages
$ virtualenv /var/lib/awx/venv/ara

# Install ARA, Ansible and required dependencies
$ /var/lib/awx/venv/ara/bin/pip install ara ansible psutil python-memcached

# Print location of the ARA callback
$ /var/lib/awx/venv/ara/bin/python -m ara.setup.callback_plugins
/var/lib/awx/venv/ara/lib/python2.7/site-packages/ara/plugins/callbacks
```

- Instance Groups
- Applications
- Settings**



**ANSIBLE
TOWER**
by Red Hat®



Dashboard
Jobs
Schedules
My View

RESOURCES

Templates

Credentials
Projects
Inventories
Inventory Scripts

ACCESS

Organizations
Users
Teams

TEMPLATES / run-tower-config

run-tower-config

DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES

*** NAME** run-tower-config **DESCRIPTION** Runs the tower-config.yaml playbook

*** INVENTORY** PROMPT ON LAUNCH default-inventory *** PROJECT** ansible-tower-cicd

CREDENTIAL PROMPT ON LAUNCH **FORKS** DEFAULT

*** VERBOSITY** PROMPT ON LAUNCH 0 (Normal) **JOB TAGS** PROMPT ON LAUNCH

LABELS **ANSIBLE ENVIRONMENT** /var/lib/awx/venv/ara/

Communauté



Communauté

- GitHub: <https://github.com/ansible-community/ara>
- IRC: #ara sur freenode
- Slack: [AREcordsAnsible.slack.com](https://arecordsansible.slack.com)
- Twitter: [@AREcordsAnsible](https://twitter.com/AREcordsAnsible)

Contactez moi si vous aimeriez contribuer !



Merci !

Questions ?

- GitHub: <https://github.com/ansible-community/ara>
- IRC: #ara sur freenode
- Slack: ARecordsAnsible.slack.com
- Twitter: [@ARecordsAnsible](https://twitter.com/ARecordsAnsible)
- Docs: ara.readthedocs.org