Simple (but useful) Ansible reporting with ara

FOSDEM 2022

- David Moreau-Simard (@dmsimard)
Parrots are smart dinosaurs
Did you know that ara had an unofficial mascot? His name is Yoshi :}

[Image of a parrot]
Ara (bird)

Ara is a neotropical genus of macaws with eight extant species and at least two extinct species. The genus name Ara is derived from the Tupi word ara, an onomatopoeia of the sound a macaw makes.

The Ara macaws are large striking parrots with long tails, long narrow wings and vividly coloured plumage. They all have a characteristic bare face patch around the eyes. Males and females have similar plumage. Many of its members are popular in the pet trade, and bird smuggling is a threat to several species.

Contents

1 Taxonomy
2 Morphology and appearance

Scientific classification

Kingdom: Animalia
Phylum: Chordata
Class: Aves
Order: Psittaciformes
Family: Psittacidae
Tribe: Arini
Genus: Ara
$ whoami

---

- **name**: David Moreau-Simard
  - **hosts**:
    - dmsimard.matrix.org
    - twitter.com/dmsimard
  - **vars**:
    - **location**: Montreal, Canada
    - **profile**: sysadmin, dev/ops, CI/CD, SRE
  - **roles**:
    - Principal Software Engineer in the Ansible community team @ Red Hat
    - Ansible user and contributor since version 1.8 or so (2014+)
    - Release the 'ansible' package in collaboration with the community working group
    - Improve tooling, workflows and processes for users, contributors and maintainers alike
Once upon a time...
What was challenging

- Thousands of CI jobs across hundreds of projects
- Spread across multiple instances of jenkins and Zuul
- Searching through thousands of lines of console
- What's the problem? Is it related to my patch?
- Sharing specific tasks with others
- Onboarding new contributors
What we tried

- Increasing verbosity (ansible-playbook -vvv)
- Enabling the `profile_tasks` callback
- Adding the `human_log` callback (pretty printing JSON)
- A callback that wrote results in HTML tables
Ansible console output
Ansible console output (verbose)
Ansible console output (very verbose)
Spoiler

Change 463517 - Merged

Remove verbose Ansible logging (use ARA instead)

Currently all role playbook executions use very verbose logging in order to help with diagnosing problems in their execution.

ARA provides the most useful information available in the verbose log output, so instead of trawling very large logs we should instead use ARA's easier to navigate output.

Change-Id: Idea3c01f7b72c73fdaa5f95c24421f4d3fa28

Author: Jesse Pretorius <jesse.pretorius@rackspace.co.uk> May 9, 2017 6:20 AM
Committer: Jesse Pretorius <jesse.pretorius@rackspace.co.uk> May 9, 2017 6:20 AM
Commit: 91df4bbb643c72964af859fa0a0ad24e3d5d4ab9c (gitweb)
Parents: 327d1346327323279bd570daaba3fe772666e5e243 (gitweb)
A bit less verbose

```bash
Before:
7307
56411

After:
392k
4255
```
ARA Records Ansible playbooks and makes them easier to understand and troubleshoot.

It's another recursive acronym.
How does it work?

How to draw an ARA

1. Draw some circles
2. Draw the rest of the ARA

POLYGONS
Recording workflow
Ansible callback plugins

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):

# [...]
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)
Getting started

Offline with the default sqlite backend

# Install Ansible and ARA (with API server dependencies) for the current user
```bash
python3 -m pip install --user ansible "ara[server]"
```

# Configure Ansible to use the ARA callback plugin
```bash
export ANSIBLE_CALLBACK_PLUGINS="$(python3 -m ara.setup.callback_plugins)"
```

# Run an Ansible playbook
```bash
ansible-playbook playbook.yaml
```

# Use the CLI to see recorded playbooks
```bash
ara playbook list
```

# Start the built-in development server to browse recorded results
```bash
ara-manage runserver
```
Getting started

With a containerized API server

```bash
# Create a directory for a volume to store settings and a sqlite database
mkdir -p ~/.ara/server

# Start an API server with podman from the image on Quay.io:
podman run --detach --tty --name ara
   --volume ~/.ara/server:/opt/ara:z
   -p 8000:8000
   quay.io/recordsansible/ara-api:latest

# or with Docker and Docker Hub:
docker run --detach --tty --name ara
   --volume ~/.ara/server:/opt/ara:z
   -p 8000:8000
   docker.io/recordsansible/ara-api:latest
```
Getting started

Configuring the callback to send data to our server

# Install Ansible and ARA (without API server dependencies) for the current user
python3 -m pip install --user ansible ara

# Configure Ansible to use the ARA callback plugin
export ANSIBLE_CALLBACK_PLUGINS="$(python3 -m ara.setup.callback_plugins)"

# Set up the ARA callback to know where the API server is located
export ARA_API_CLIENT="http"
export ARA_API_SERVER="http://127.0.0.1:8000"

# Run an Ansible playbook
ansible-playbook playbook.yaml

# Browse http://127.0.0.1:8000 to view the reporting interface
Live demo

Recording an Ansible playbook from AWX

1. Deploy a kubernetes node with k3s-ansible
2. Run an ara server to receive results from AWX
3. Deploy AWX with awx-operator
4. Configure AWX to record results with ara
5. Run a playbook with AWX and see if it works
More reading

- Recording Ansible playbooks from AWX with ara
Another live demo

demo.recordsansible.org

Deployed with the ara Ansible collection with gunicorn, nginx and mariadb
The project could use your help

- https://github.com/ansible-community/ara
- https://github.com/ansible-community/ara-collection
- https://github.com/ansible-community/ara-infra
- https://github.com/ansible-community/ara-web (broken)
- https://ara.readthedocs.org
Thank You!

Any questions?

Stay up to date and come chat with us:

- https://ara.recordsansible.org
- @RecordsAnsible on Twitter
- https://matrix.to/#/#ara:libera.chat
- #ara on libera.chat IRC network

Link to these slides (and more): https://ansible.github.io/community/decks/