

Ansible Galaxy

Serena Lorenzini (*BioDec.com*)

Pycon otto

- What is Ansible?
- What is Ansible Galaxy?
- Download and advanced download.
- Create reusable Ansible roles with the Galaxy.
- Ansible, Github and TravisCI.
- The Biodec role template.

What is 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 only SSH (and python).
- No agents to install on remote systems.

What is Ansible?

Ansible is a python tool that automates the provisioning of an IT infrastructure through SSH connection.

Ansible instructions are written in plain yaml and can (have to) be organized in *roles*.

Why should I use it?

Because you are doing at least one of these activities:

- Networking
- Software deployment
- Software upgrades
- Security upgrades
- Provisioning systems
- User management in OS
- Building clusters

Install Ansible

Being a python product installing Ansible is as easy as typing:

```
$ pip install ansible
```

- Tip: install Ansible (preferably) in a dedicated python virtualenv using the pip python package manager.

```
$ pip install virtualenv
```

```
$ virtualenv myproject
```

```
$ . myproject/bin/activate
```

```
$ (myproject) pip install Ansible==2.1.4.0
```

Install Ansible

and you will find a load of ansible-related tools:

ansible

ansible-console

ansible-container

ansible-doc

ansible-galaxy

ansible-lint

ansible-playbook

ansible-pull

ansible-vault

What is Ansible Galaxy?

It is the Ansible's official community hub for finding, downloading, rating, and sharing Ansible roles..."

So, it is ...

- a command line tool
- a website

for searching, installing, creating and managing roles.

Many many projects can be found in the galaxy

- Many many duplicates
- Quality from *excellent* through *broken* to *horribly dangerous*

A word on roles

Roles are a level of abstraction that allows to write clean and reusable ansible code.

Roles are just automation around `include` directives, and don't contain much beyond some improvements to search path handling for referenced files. However, that can be a big thing!

Why using Ansible Galaxy?

Ansible is easy to learn but rushing to use it with a limited knowledge of its best practices leads to not reusable code.

Creating reusable Ansible roles is a strongly encouraged practice.

During this presentation we will take a look at a tool designed for this purpose: Ansible Galaxy. It comes bundled with Ansible and its main purpose is to share roles using the Galaxy hub and other SCMs.

Install Ansible Galaxy

Ansible Galaxy comes bundled with Ansible. Install Ansible and type your first Galaxy command:

```
$ ansible-galaxy -h  
Usage: ansible-galaxy
```

```
delete  
import  
info  
init  
install  
list  
login  
remove  
search  
setup
```

Download a role from the Galaxy hub

Let's look at some useful command: install.

```
$ ansible-galaxy install username.rolename
```

And it will download the role from

<https://galaxy.ansible.com/username/rolename/> in the ansible roles path.

Advanced download

You can do more than installing one role at time...

- Install multiple roles at once.
- Install roles from various SCMs.
- Specify the roles path.

Install multiple roles at once

Write a `requirements.yml` file. It contains a list of roles to be downloaded:

```
# from galaxy
- src: userone.roleone
- src: usertwo.roleone
- src: usertwo.roletwo
```

Provide the file to `ansible-galaxy` command:

```
$ ansible-galaxy install -r requirements.yml
```

Install roles from various SCM

Be sure to have set the credentials (i.e. add SSH key) to authenticate yourself.

- From Github:

```
# from Github
- src: https://github.com/bennojoy/nginx
  version: master
  name: nginx_role
```

Install roles from various SCM

- From a webserver:

```
# where the role is packaged in a tar.gz
- src: https://webserver.example.com/files/master.tar.gz
  name: http-role
```

- From other git scm:

```
# from GitLab or other git-based scm
- src: git@gitlab.company.com:mygroup/ansible-base.git
  scm: git
  version: "0.1" # quoted, for YAML parser
```


Specify the roles path

Ansible downloads roles to the path specified by the environment variable `ANSIBLE_ROLES_PATH`. This can be set to a series of directories (i.e. `/etc/ansible/roles:~/.ansible/roles`).

You can override this by:

- setting the environment variable in your session
- defining `roles_path` in an `ansible.cfg` file
- using the `-roles-path` option.

```
$ ansible-galaxy install --roles-path . -r requirements.yml
```

Create Ansible roles using Ansible Galaxy

The `init` command

The first step in creating a role is creating its directory structure.

Use the `init` command to initialize the base structure of a new role, saving time on creating the various directories and `main.yml` files a role requires.

```
$ ansible-galaxy --offline init role_name
```

The above will create a role named *role-name*.

Create Ansible roles using Ansible Galaxy

The role structure:

```
README.md
.travis.yml
defaults/main.yml
files/
handlers/main.yml
meta/main.yml
templates/
tests/
    inventory
    test.yml
vars/main.yml
```

...wait... a Travis file?

Ansible and Travis

- Github is the main git scm for sharing Ansible roles, and it integrates very well with TravisCI.
- Starting from ansible 2 the role template contains a minimal setup for running tests with Travis.

Ansible and Travis

The travis file:

```
language: python
python: "2.7"
```

```
addons:
```

```
  apt:
```

```
    packages:
```

```
      - python-pip
```

```
install:
```

```
  - pip install ansible
```

```
script:
```

```
  - ansible-playbook tests/test.yml -i tests/inventory \
--syntax-check
```

Ansible galaxy, github and Travis

The galaxy imports roles from Github and, since Ansible 2, it shows the Travis build status as well!

You will need:

- the Travis token to authenticate your repo `ansible-galaxy` setup
- to add notifications to `travis.yml` file

`notifications:`

```
webhooks: https://galaxy.ansible.com/api/v1/notifications/
```

What if I have roles on another scm?

Like we do in Biodec? We save our work in a self-hosted Gitlab service. Gitlab is a great alternative to Github and has its own services for running CI/CD pipelines.

The Biodec role template

We have come to like a different role structure when developing our roles, mainly because we have our private Gitlab server where we push/download our roles instead of Github.

- We have a `gitlab-ci.yml` (not `travis.yml`).
- We have a `requirements.txt` for installing ansible via pip (we like to fix the versions).
- We have a `ansible.cfg` to declare roles path (instead of writing it on the fly).
- We have a `install_roles.yml` to download other roles, along with `meta.yml`.
- We don't have `tests dir` (there are no proper tests). and we have moved `test.yml` in `project dir`.

The Biodec role template

```
.gitlab-ci.yml  
README.md  
ansible.cfg  
defaults/main.yml  
files/  
install_roles.yml  
handlers/main.yml  
meta/main.yml  
templates/  
test.yml  
production.yml  
requirements.txt  
vars/main.yml
```

Our CI file

CI tests in Docker containers:

```
test:
```

```
  image: python:2.7
```

```
  script:
```

```
    - apt-get update -y && \
      apt-get install -y python python-dev python-pip
    - pip install -r requirements.txt
    - echo localhost > inventory
    - ansible-playbook -i inventory \
      test.yml --connection=local
```

Creating roles using our template

How do you create an Ansible-galaxy template?

The “easy way”:

```
ansible-galaxy init --role-skeleton=/path/to/skeleton role_name
```

Or configure `ansible.cfg`

```
[galaxy]
role_skeleton = /path/to/skeleton
role_skeleton_ignore = ^.git$,^.*/.git_keep$
```

Creating roles using our template

Other ways:

- Fork the code (not easily updatable) so the `init` command will load a different role template.
- Create a git project where you put the role template and run `git clone` instead of `ansible-galaxy init`.

Conclusion

- Ansible is easy to start
- Ansible solves 80% of our problems (devops people)
- Ansible is a *popular* technology constantly evolving and adding modules
- it keeps up with almost every type of OS or virtualization type
- You can use Ansible Galaxy with your *private* galaxy of roles and you can build your own role skeleton.

4devops

4devops next appointments

- Workshop Docker e Container day, 27-28 Aprile 2017, Verona (Italy)
- Progettare e far evolvere infrastrutture cloud su AWS, 4 maggio 2017, Milano (Italy)
- Orchestrators and Containers day, 24 May 2017 in Chiasso (Switzerland)

Thank you

Contacts

Serena Lorenzini email: serena@biodec.com

BioDec www.biodec.com