Эффективная организация параметров групп и хостов в Ansible

Serge Matveenko
DataArt

github.com/lig

Ansible?

- Provisioning
- Configuration Management
- Application Deployment
- Continuous Delivery
- Security & Compliance
- Orchestration
- …

www.ansible.com

What's the catch?

- Playbooks become messy
- A lot of hosts
- A lot of roles
- A lot of vars
- Something to think about

What the docs say

```
roles/
        common/
                files/
                templates/
                tasks/
                handlers/
                vars/
                defaults/
                meta/
        webservers/
                …
        scrappers/
                …
        …
```

```
group_vars/
        group1          # here we assign variables to particular groups
        group2          # ""
host_vars/
        hostname1          # if systems need specific variables, put them here
        hostname2          # ""
site.yml          # master playbook
webservers.yml          # playbook for webserver tier
dbservers.yml          # playbook for dbserver tier
roles/
        common/
        webtier/
        monitoring/
        fooapp/
```
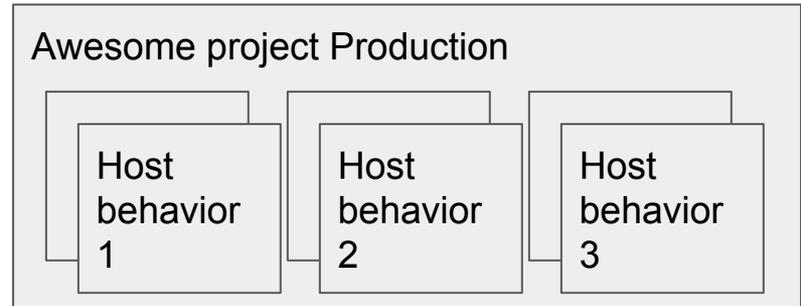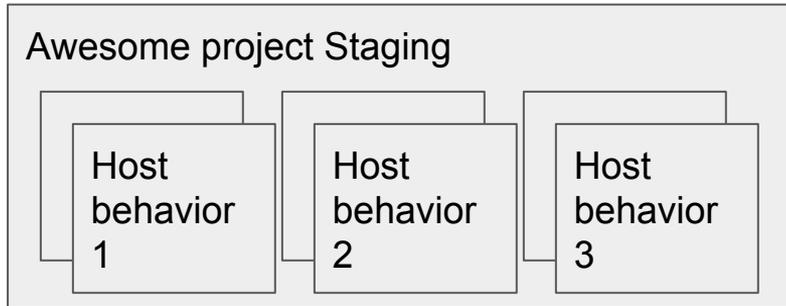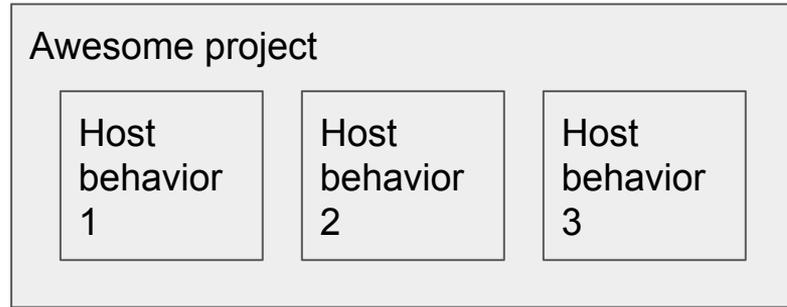
Real world

- Different environments (production, staging, development, experimental)
- 20+ host behaviors
- 100+ hosts
- Automatic discovery
- Pre-/post-play host tasks
- …

Approach Requirements

- Environment configuration
- Host behavior configuration
- Custom scenarios in some cases
- DRY (Don't Repeat Yourself)
- Extensibility
- Easy understandable codebase
- … all the good things we love 🤍

How it looks

Awesome project

| Host behavior 1 | Host behavior 2 | Host behavior 3 |

Awesome project Staging

| Host behavior 1 | Host behavior 2 | Host behavior 3 |

Awesome project Production

| Host behavior 1 | Host behavior 2 | Host behavior 3 |

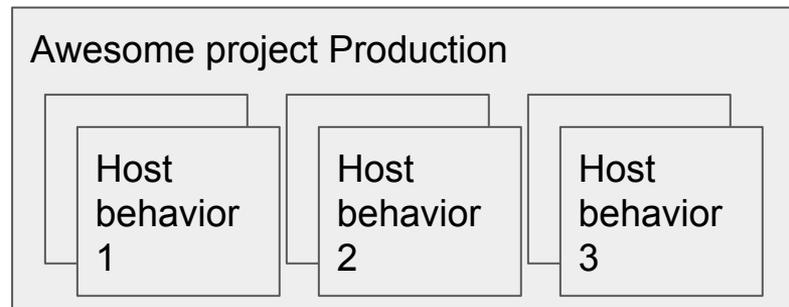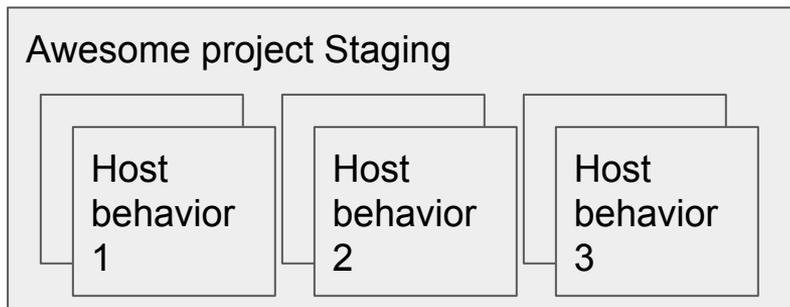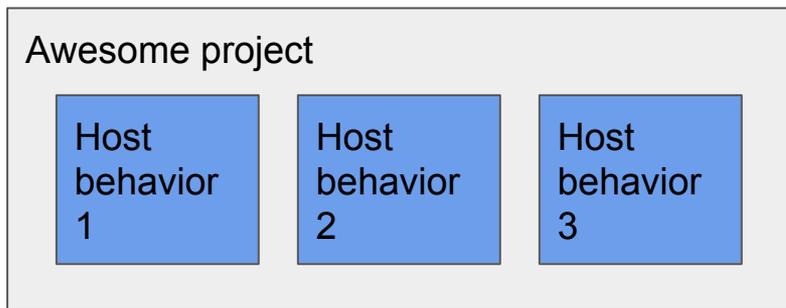# A dumb way

- Staging
  - Staging Host behavior 1
    - Staging Host 1.1
    - Staging Host 1.2
  - Staging Host behavior 2
    - Staging Host 2.1
    - Staging Host 2.2
- Production
  - Production Host behavior 1
    - Production Host 1.1
    - Production Host 1.2
  - Production Host behavior 2
    - Production Host 2.1
    - Production Host 2.2

[staging:children]
staging-behavior-1
staging-behavior-2

[staging-behavior-1]
staging-host-1-1
staging-host-1-2

[staging-behavior-2]
staging-host-2-1
staging-host-2-2

How it looks

Awesome project

Host behavior 1 | Host behavior 2 | Host behavior 3

Awesome project Staging

Host behavior 1 | Host behavior 2 | Host behavior 3

Awesome project Production

Host behavior 1 | Host behavior 2 | Host behavior 3

Green & Red

It's just beautiful …

The Solution

Green   Red

- Project wide configuration
- Behavior specific configuration
- Special scenarios tasks

- Contain green and red groups
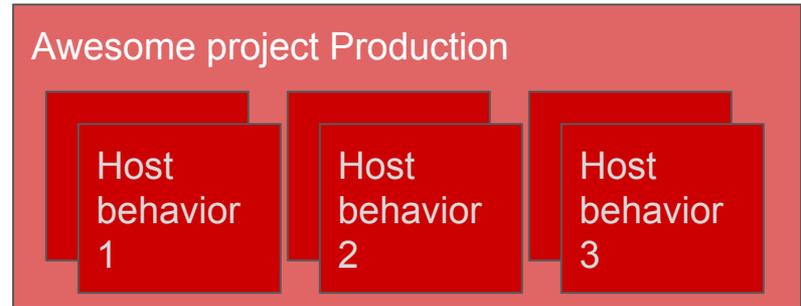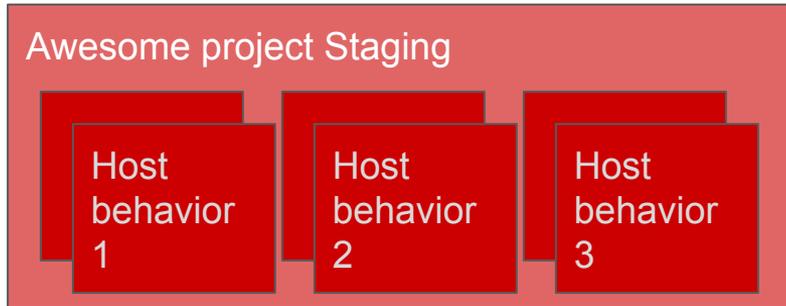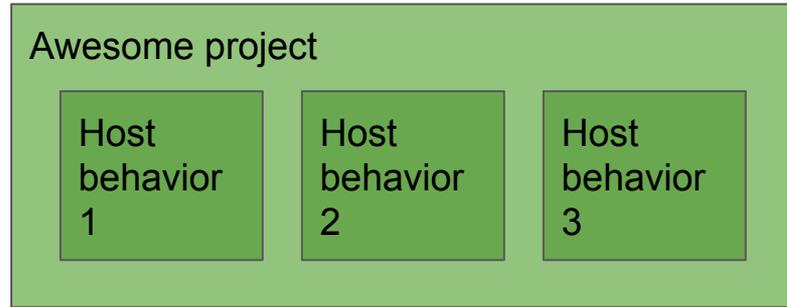- Contain green vars only

- Act like classes

- Environment specific configuration
- Host specific configuration
- Do not appear in playbooks

- Contain red groups only
- Contain red vars only

- Act like instances

Let's add the colors

Awesome project

| Host behavior 1 | Host behavior 2 | Host behavior 3 |

Awesome project Staging

| Host behavior 1 | Host behavior 2 | Host behavior 3 |

Awesome project Production

| Host behavior 1 | Host behavior 2 | Host behavior 3 |

The right way

[project:children]
staging
production

[staging:children]
staging-behavior-1
staging-behavior-2

[behavior-1:children]
staging-behavior-1
production-behavior-1

[behavior-2:children]
staging-behavior-2
production-behavior-2

[staging-behavior-1]
staging-host-1-1
staging-host-1-2

[staging-behavior-2]
staging-host-2-1
staging-host-2-2

[custom-scenario:children]
behavior-2

The right way

[project:children]
staging
production

[staging:children]
staging-behavior-1
staging-behavior-2

[behavior-1:children]
staging-behavior-1
production-behavior-1

[behavior-2:children]
staging-behavior-2
production-behavior-2

[staging-behavior-1]
staging-host-1-1
staging-host-1-2

[staging-behavior-2]
staging-host-2-1
staging-host-2-2

[custom-scenario:children]
behavior-2

Thank you!

github.com/lig