Ansible Basics

Be smart. Think open source.

Ansible - Basics

Simple config management and orchestration

```
- name: teach ansible basics
  template:
    src: good_ideas.j2
    dest: customers
    state: present
```

Agenda

- Config management vs. orchestration
- Introduction to Ansible
- Basic components
- Variables
- Templates

Config management vs. orchestration

Learn the difference

Config management

> "Define how a system should look like in an abstract way."

- Ensure Apache and MariaDB is installed
- Ensure file /etc/motd contains line XYZ

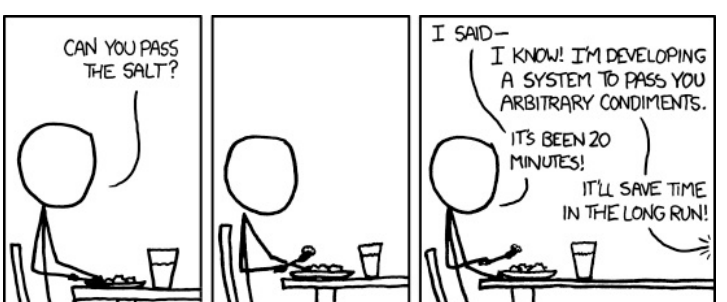
Orchestration

> "Run a set of tasks on a set of servers at once."

- Update and restart exactly 49% of a clustered service to ensure the quorum
- Patch all systems which are vulnerable to the Dirty COW bug

Goals

- Reproducibility is key
- Consistency
- Save time in the long run*



Introduction to Ansible

What's it all about?

Facts

- Project started in 2012
- Licensed under the GPLv3
- Acquired by Red Hat in October 2015
- Development pushed by Red Hat and community is growing fast

Ecosystem

- Ansible
- Ansible AWX / Tower
- Ansible Container (the new face in town)
- Ansible Galaxy

Design

- Agentless
- YAML based configuration via SSH
- Written in Python 2
- Template rendering with Jinja2

Strong suite

- Simplicity is key
- Easy as pie
- Zero configuration (almost)
- Works via SSH

Strong suite

- Idempotence
- Small footprint

Weak spots

- Does not scale as well as other tools
- Certain complex tasks turn ugly

Weak spots

- Ansible Galaxy (proceed with caution!)
- Contributions are somewhat slowly processed

Ansible components

Basic terminology you need to know

Modules

Modules are used to interact with nodes and manage different resources:

- ping
- command
- copy / synchronize
- lineinfile
- package
- mysql_user / mysql_db

Have a look at the Ansible Module Index!

Modules

Modules can be executed ad-hoc:

```
$ ansible web -i inventory.txt -u root -m ping
$ ansible web -i inventory.txt -u root -m command -a "df -h"
```

Each module exposes different options that can be customized

Inventory

An inventory contains all the target nodes and structures them into groups:

```
[adfinis:children]
bern
basel

[bern:children]
database

[basel:children]
web

[database]
db[01:03].adfinis-sygroup.ch ansible_user=db_admin

[web]
web[01:03].adfinis-sygroup.ch ansible_user=web_admin
```

Hands-on :: Basics 01

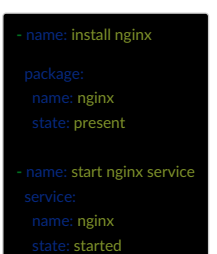
Install Ansible and take the first steps

Ansible components 2

Further down the rabbit hole

Tasks

- Tasks are Ansible commands which call a module
- Written in a YAML file
- Executed from top to bottom



Handlers

- Handlers are basically tasks but notified by other tasks
- Executed only when necessary

```
- name: restart nginx
  service:
    name: nginx
    state: restarted
```

Playbooks

Group tasks and handlers together in a playbook and make them reusable

```yaml
---
- hosts: web
  tasks:
    - name: template nginx.conf
      template:
        src: nginx.conf.j2
        dest: /etc/nginx/nginx.conf
        backup: yes
      notify:
        - restart nginx
  handlers:
    - name: restart nginx
      service:
        name: nginx
        state: restarted
```

Playbooks

Execute the playbook:

```
$ ansible-playbook webserver.yml -i inventory.txt -u root
```

Roles

Divide and structure a playbook into different roles:

- Each role is responsible for a certain component
- Reuse the same roles in many different projects (playbooks)

```
---
- hosts: web
  roles:
    - common
    - webserver
    - monitoring
```

Roles

A role has a specific directory structure:

```
ansible
|-- webserver.yml
|-- roles
    |-- nginx
        |-- defaults
        |-- files
        |-- handlers
        |-- tasks

        |-- templates
        |-- vars
        |-- meta
```

Facts

Node specific information stored in variables:

- Hardware resources
- Network configuration
- Operating system
- and much much more

Facts

Use the `setup` module to gather and display facts:

```
$ ansible web -i inventory.txt -u root -m setup
```


```
"ansible_facts": {
    "ansible_all_ipv4_addresses": [
        "192.168.122.10"
    ],
    "ansible_all_ipv6_addresses": [
        "fe80::5054:ff:fe5c:593"
    ],
    "ansible_architecture": "x86_64",
    "ansible_bios_date": "04/01/2014",
    "ansible_bios_version": "1.9.3-20160701_074356-anatol",
[...]
```

Hands-on :: Basics 02

Create some tasks and the first playbook

Variables

How to make your playbooks adaptable

General overview

- Ansible allows one to use variables instead of hard coded values
- Variables can be overwritten and included from different places
- Support for different types (strings, numbers, lists, etc.)
- Facts are variables too

General overview

- Example of several vars:

```
nginx_packages:
  - nginx

nginx_conf_dir: /etc/nginx/conf.d


nginx_server_name: "{{ ansible_fqdn }}"
```

- Variables used within a task:

```
- name: install nginx
  package:
    name="{{ nginx_packages }}"
    state=present
```

Scope

Ansible has 3 different variable scopes:

- Global (config, ENV & commandline)
- Play (vars, include_vars, role defaults & vars)
- Host (specific to a machine like facts)

Locations

Variables can be included from many different locations:

- Inventory
- group_vars
- host_vars
- Playbook
- Roles defaults
- Roles vars

group_vars

Group variables are included based on the node's groups:

```
[adfinis:children]
bern
basel

[bern:children]
database

[basel:children]
web

[database]
db[01:03].adfinis-sygroup.ch ansible_user=db_admin

[web]
web[01:03].adfinis-sygroup.ch ansible_user=web_admin
```

host_vars

Create host specific variables in the directory host_vars:

- ansible/host_vars/db01.adfinis-sygroup.ch
- ansible/host_vars/web01.adfinis-sygroup.ch

Play vars

It's possible to include vars in your playbooks:

```
---
- hosts: web
  vars:

    nginx_package: nginx
  vars_files:
    - /vars/external_vars.yml
  roles:
    - nginx
```

Role variables

To define vars in your role create the following files:

- roles/nginx/defaults/main.yml
- roles/nginx/vars/main.yml

Prefix all variables with the role name to prevent conflicts!

Variable precedence

Ansible has a tricky variable precedence:

- role defaults
- inventory
- playbook
- host facts
- play vars
- registered vars
- role and include vars
- extra vars

See the documentation!

How you should use them

Start simple and restrict yourself to the following vars:

- role defaults
- group_vars
- host_vars
- role vars

Hands-on :: Basics 03

Make your playbook more dynamic with variables

Templates

Generate configurations on the fly

General overview

Ansible supports rendering of templates through Jinja2:

- Use the template module
- Create a file in the role directory "templates" with the suffix .j2

```
server {
    listen       80 default_server;
    server_name  {{ nginx_server_name }};

    location / {
        root /var/www;
    }
}
```

Loops

Jinja2 templates also support loops to generate multiple config blocks:

```yaml
nginx_locations:
  - path: /web
    alias: /var/www/web
  - path: /admin
    alias: /var/www/admin
```

```
{% for item in nginx_locations %}
    location {{ item.path }} {
        alias {{ item.alias }};
        autoindex on;
    }

{% endfor %}
```

Conditions

Use conditions (if, else, etc.) to have even more control:

```
{%- if nginx_ssl %}
server {
    listen       443 default_server;
    server_name  {{ nginx_server_name }};
[...]
{%- endif %}
```

Hands-on :: Basics 04

Generate files dynamically with templates

Field report

What have you learned?

- Ansible basics
- Ansible components
- Variable handling
- Template rendering

Field report

Can you describe all directories?

Quo vadis?

- Ansible Best Practice
- Ansible Vault
- Multi distribution support
- Development Workflow (CI & CT)
- Roll out strategy and cluster management

Feedback

The good, the bad and the ugly

Thank you!

Be smart. Think open source.

Feel Free to Contact Us

www.adfinis-sygroup.ch

Tech Blog

GitHub

info@adfinis-sygroup.ch

Twitter

Attribution / License

- XKCD - The General Problem, by xkcd https://xkcd.com/974/ License CC-BY-NC see https://xkcd.com/license.html