Ansible Basics

Be smart. Think open source.

Ansible Hands-on

Learning by doing

Hands-on :: Basics 01

Install Ansible and take the first steps

Basics 01 - Installation

Install Ansible on your machine:

- RHEL & CentOS (requires EPEL)

```
$ sudo yum install ansible
```

- Debian & Ubuntu

```
$ sudo apt-get install ansible
```

Basics 01 - Installation

Check if you have the latest Ansible version:

```
$ ansible --version
$ man ansible
```

Basics 01 - Installation

Add your SSH public key to the authorized_keys file on the target node:

```
$ ssh-copy-id root@192.168.122.10
```

Basics 01 - Installation

Create a working directory for this workshop:

```
$ mkdir ~/ansible_workshop
$ cd ~/ansible_workshop
```

Basics 01 - Inventory

Create the file inventory.txt containing your test node:

```
[test]
192.168.122.10
```

Basics 01 - Ad-hoc commands

Execute your first ad-hoc commands:

```
$ ansible test -i inventory.txt -u root -m ping
$ ansible test -i inventory.txt -u root -m command -a "df -h"
$ ansible test -i inventory.txt -u root -m command -a "ls -l /"
```

Hands-on :: Basics 02

Create some tasks and the first playbook

Basics 02 - Facts

Explore the facts of your test node:

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

Basics 02 - Playbooks

Create the file webserver.yml with the following content:

```
---
- hosts: test
  tasks:

    - name: install nginx
      package:
        name: nginx
        state: present

    - name: start nginx service
      service:
        name: nginx
        state: started
```

Basics 02 - Playbooks

Run the playbook against your test node:

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

Was it successful? Check if the webserver is running in your browser!

Basics 02 - Playbooks

Get debugging output by adding the verbose flag(s):

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

Add more -v parameters to get even more output

Basics 02 - Roles

Create a new role called "nginx":

```
$ mkdir -p roles/nginx/tasks
```

Add the previous tasks to the tasks file roles/nginx/tasks/main.yml:

```
---
- name: install nginx
  package:
    name: nginx
    state: present

- name: start nginx service
  service:
    name: nginx
    state: started
```

Basics 02 - Roles

Include the new nginx role in the webserver.yml playbook:

Execute the playbook again, what happens?

Hands-on :: Basics 03

Make your playbook more dynamic with variables

Basics 03 - Variables

Create a host_vars and group_vars directory in your working dir:

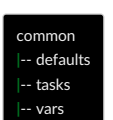
```
$ mkdir host_vars group_vars
```

Your directory now should look like this:

```
ansible_workshop
|-- group_vars
|-- host_vars
|-- inventory.txt
|-- roles
|-- webserver.yml
```

Basics 03 - Common role

Create an additional role called "common" including the following directories:



Basics 03 - Common role

Add the defaults vars listed below:

```
---
common_packages:
  - ntp
  - iptables
```

Basics 03 - Common role

The new role should take care of installing several packages:

```
---
- name: install common packages
  package:
    name: "{{ item }}"
    state: present
  with_items: "{{ common_packages }}"
```

Basics 03 - Testing

Include the new role into your playbook and give it a spin:

- What happens?
- What packages were installed?

Basics 03 - group_vars

Create the file group_vars/test with the following variables:

```
---
common_packages:
  - ntp
  - iptables
  - vim
  - zsh
```

Basics 03 - Testing

Execute the playbook a second time:

- Are there any changes?
- If yes, why?

Basics 03 - Role vars

Create the file roles/common/vars/main.yml with the following variables:

```yaml
---
common_packages:
  - ntp
  - iptables
  - vim
  - zsh
  - tcpdump
  - wget
  - curl
  - rsync
```

Basics 03 - Testing

Execute the playbook a third time:

- Wait, what happened now?
- Please explain to me!

Hands-on :: Basics 04

Generate files dynamically with templates

Basics 04 - Preparation

Add the missing directories in our nginx role:

- defaults
- handlers
- templates
- vars

Basics 04 - Defaults

Add the following variable to the defaults vars:

```
---
nginx_welcome_messages:
  - "Ansible is cool!"
  - "It even gets better!"
  - "My first loop!"
```

Basics 04 - Template

Create the new template called `index.html.j2`:

```
{% for message in nginx_welcome_messages %}
<p>{{ message }}</p>
{% endfor %}
```

Basics 04 - Tasks

Which module do we need to render the template and copy it to /usr/share/nginx/www?

- Add a new task which renders the template
- Create a backup of the old index.html file

Basics 04 - Testing

Execute your modified playbook to deploy the new website:

- Did it work?
- Which messages are displayed?

Basics 04 - Handlers

Imagine this is a more complex web application:

- Restart the nginx service if the index.html file changed

Basics 04 - Testing

Deploy the website again:

- Did it work?
- Was the nginx service restarted?

Good work!

You've completed this part of the workshop!

Feel Free to Contact Us

www.adfinis-sygroup.ch

Tech Blog

GitHub

info@adfinis-sygroup.ch

Twitter