ANSIBLE - Essentials

eosc-hub.eu

@EOSC_eu

Doina Cristina Duma, INFN-CNAF

Marica Antonacci, INFN-Bari

"Instanziazione e utlizo batch system on demand su infrastrutture cloud. L'esempio pilota dell'esperimento AMS»

25 – 28 Nov. 2019, Perugia

EOSC-hub receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 777536.

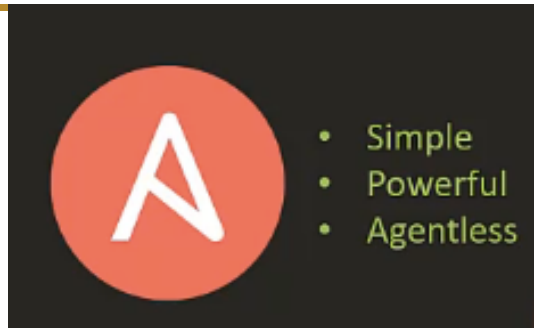
Outline

- What is, how it works, architecture
- Key components
  - Ad-hoc commands
- Roles, their structure
- Ansible-Galaxy & Galaxy, Roles use and re-use
- Playbooks & roles
- Advanced usage: debug, optimization

Bit…s of History

- «**Ansible**»
  - 1966 – **Ursula K. Le Guin**, *«Rocannon's World»*
    - *«**answerable**»: device that allow its users to **receive answers** to their **messages** in a **reasonable** amount of **time**, even over **interstellar distances***
  - 1977,1985 – **Orson Scott Card**, *«Ender's Game»*
    - *«**Philotic Parallax Instantaneous Communicator**»: machine capable of **communicating** across **infinite distances** with **no time delay***
  - **2012** - **Michael DeHaan** , RH Emerging Technlogies: ***«work on basically whatever they thought people needed»***
    - Cobbler & Func
    - AnsibleWorks, Inc. => Ansible, Inc. => RedHat (2015)
    - ***«a simple deployment, model-driven configuration management, and command execution framework»***

What is?



*«Ansible is an **automation** and **configuration** management technology used to **provision**, **deploy**, and **manage** compute infrastructure across **cloud**, **virtual**, and **physical** environments»*

- Automation language that can describe an IT application infrastructure, in Ansible Playbooks => YAML
- Automation Engine that runs Ansible Playbooks

(YAML = YAML Ain't Markup Language)

- **Human friendly (readble) data-serialization standard for all programming languages**
- Can be used with nearly any application that needs to **store** or **transmit** data
- **Flexible** = bits and pieces from **other languages**:
  - Scalars, lists, associative arays <- **Perl**
  - Document separator, «—» <- **MIME**
  - Whitespace wrapping <- **HTML**
  - Escape sequences <- **C**
  - uses both **Python**-style indentation to indicate nesting
  - Superset of **JSON** - uses [] for lists and {} for maps

YAML version

```
---
# <- yaml supports comments, json does not # did you know you can
embed json in yaml? # try uncommenting the next line # { foo: 'bar' }
json: - rigid - better for data interchange yaml: - slim and flexible
- better for configuration object: key: value array: - null_value: -
boolean: true - integer: 1 paragraph: >
   Blank lines denote

   paragraph breaks
content: |-
   Or we
   can auto
   convert line breaks
   to save space
```

JSON version

```
{
  "json": [
    "rigid",
    "better for data interchange"
  ],
  "yaml": [
    "slim and flexible",
    "better for configuration"
  ],
  "object": {
    "key": "value",
    "array": [
      {
        "null_value": null
      },
      {
        "boolean": true
      },
      {
        "integer": 1
      }
    ]
  },
  "paragraph": "Blank lines denotenparagraph breaksn",
  "content": "Or wencan autonconvert line breaksnto save space"
}
```

Ansible is …

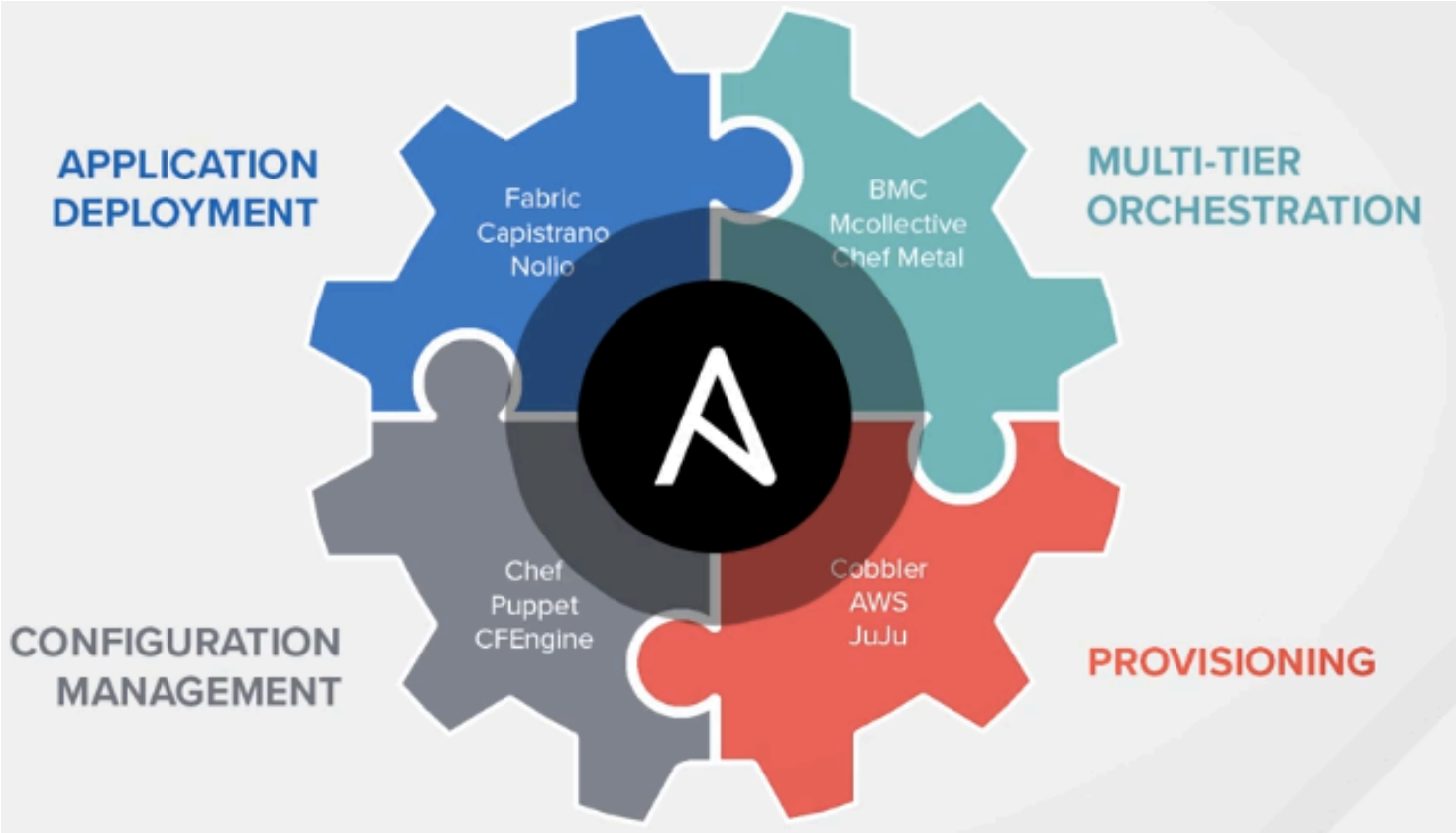
- **Simple**
  - Human readable automation
  - No special coding skills needed
  - Tasks executed in order
  - Get productive quickly
- **Powerful**
  - Application deployment
  - Configuration management
  - Workflow orchestration
  - Orchestrate the application lifecycle
- **Cross-platform**
  - Agentless support for all major OS, physical, virtual, cloud and network

- **Works with existing toolkits**
  - Homogenize existing env. By leveraging current toolsets and update mechanisms
- **«Batteries Included»**
  - Comes bundled with > 450 modules

- Cloud
- Containers
- Databases
- Files
- Messaging
- Monitoring
- Network
- Notifications
- Packaging
- Source Control
- System
- Testing
- Utilities
- Web Infrastructure

- **Community powered**
  - the **most popular open source automation** tool on **GitHub**
    - Downloads ~250k/month
    - People – 3500 people contributing modules, 1200 users on IRC

Ansible – Complete Package



Use cases

CONFIG MANAGEMENT

Enforce a baseline state for all of our servers
- right packages the right configuration files

APP DEPLOYMENT

«Take code and push it to servers»
- deploy multi-tier applications reliably and consistently, all from one common framework

PROVISIONING

Latest cloud platforms, virtualized hosts and hypervisors, network devices and bare-metal servers

CONTINUOUS DELIVERY

Push-based architecture allows very fine-grained control over operations

SECURITY & COMPLIANCE

Apply and enforce security standards that adapt to meet internal and external security guidelines

ORCHESTRATION

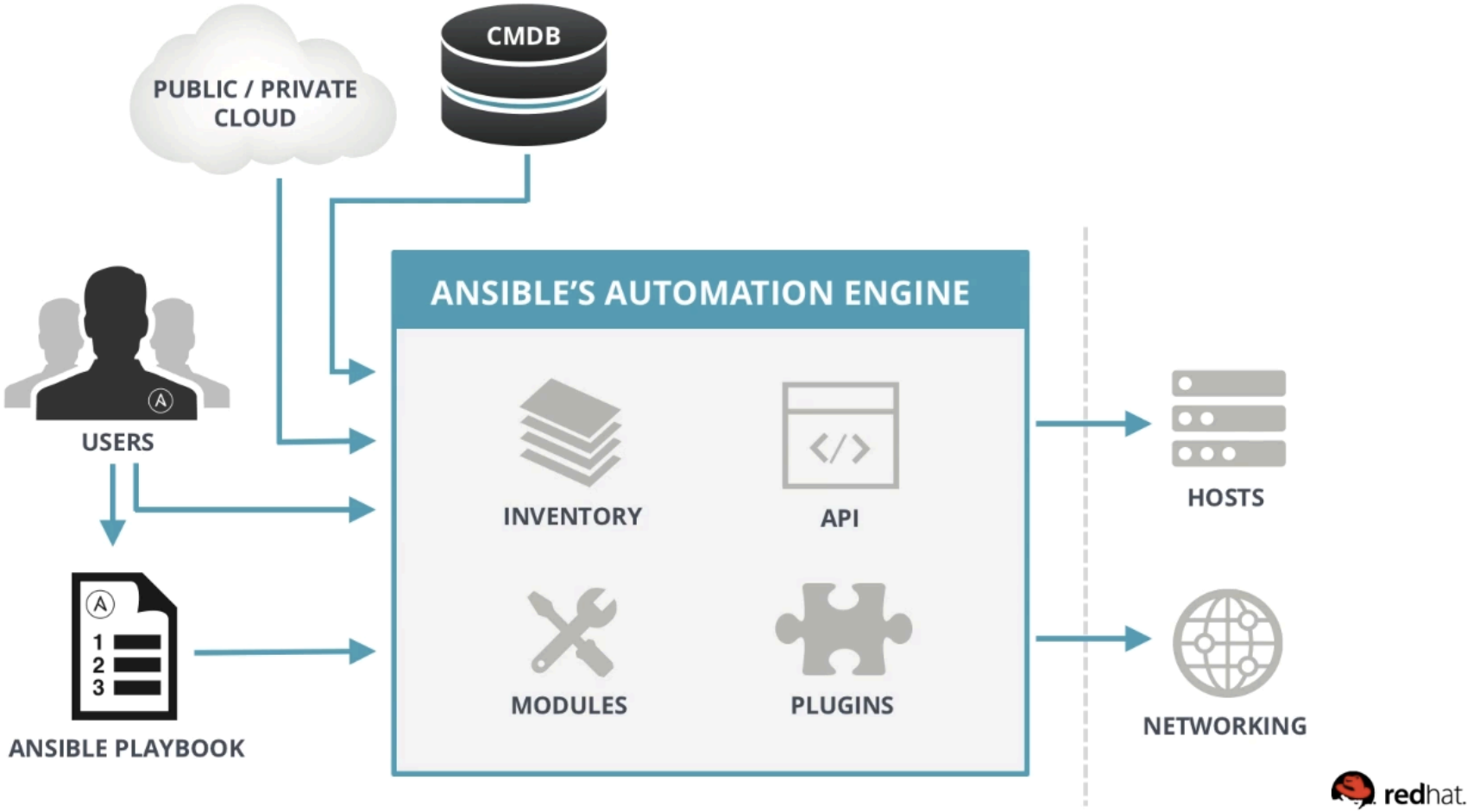
«Bring order into caos»
- targeting the right servers in the right order at any time

Ansible concepts

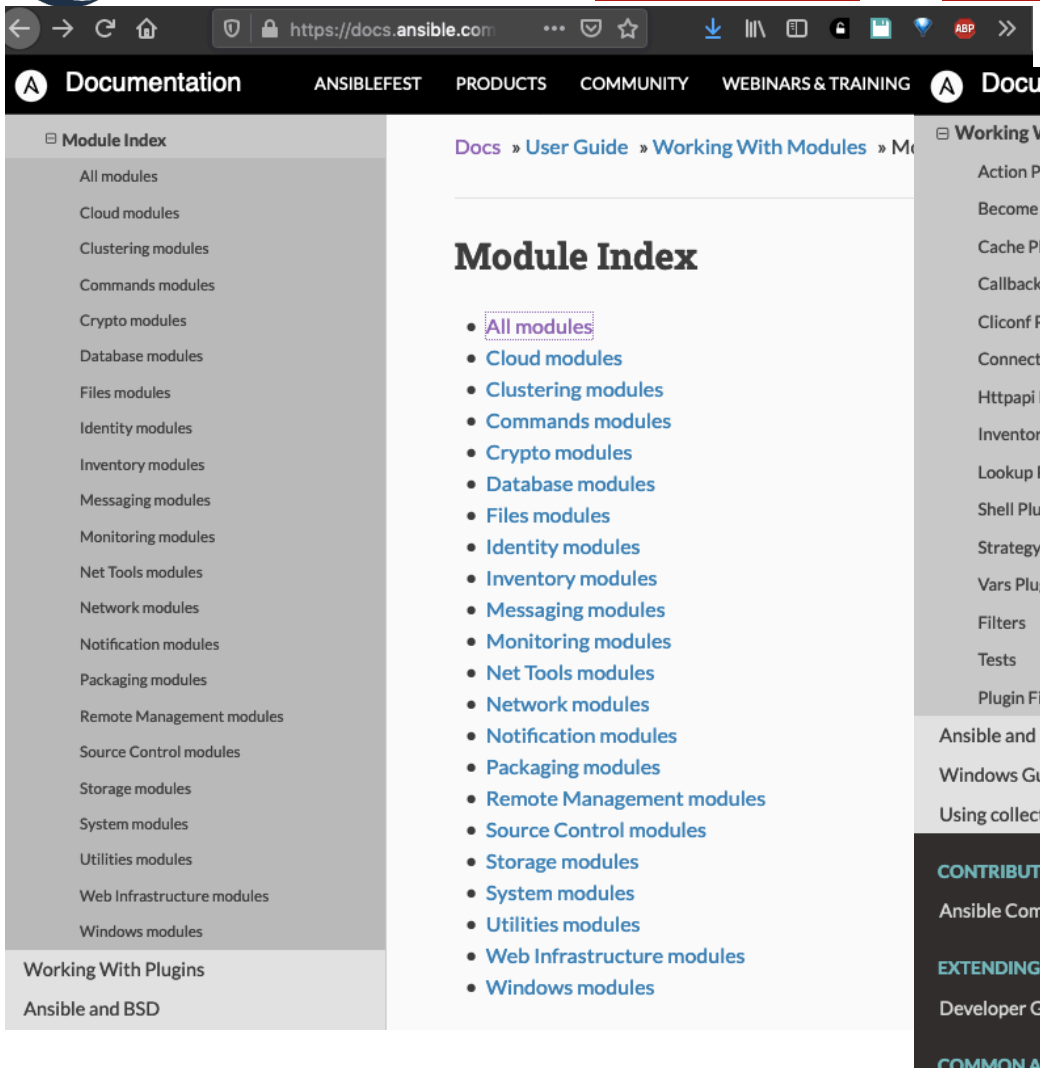
- ***Control Node***
  - Any machine with Ansible installed
- ***Managed Nodes*** = hosts
  - Servers one manages **with** Ansible
  - No Ansible installed
- ***Inventory*** = hostfile
  - List of managed hosts
  - Groups – hosts with common features (web server, rack)
- ***Modules***
  - units of code Ansible executes
- ***Tasks***
  - units of action in Ansible
- ***Playbook***
  - Ordered lists of tasks, and variables
  - Written in YAML

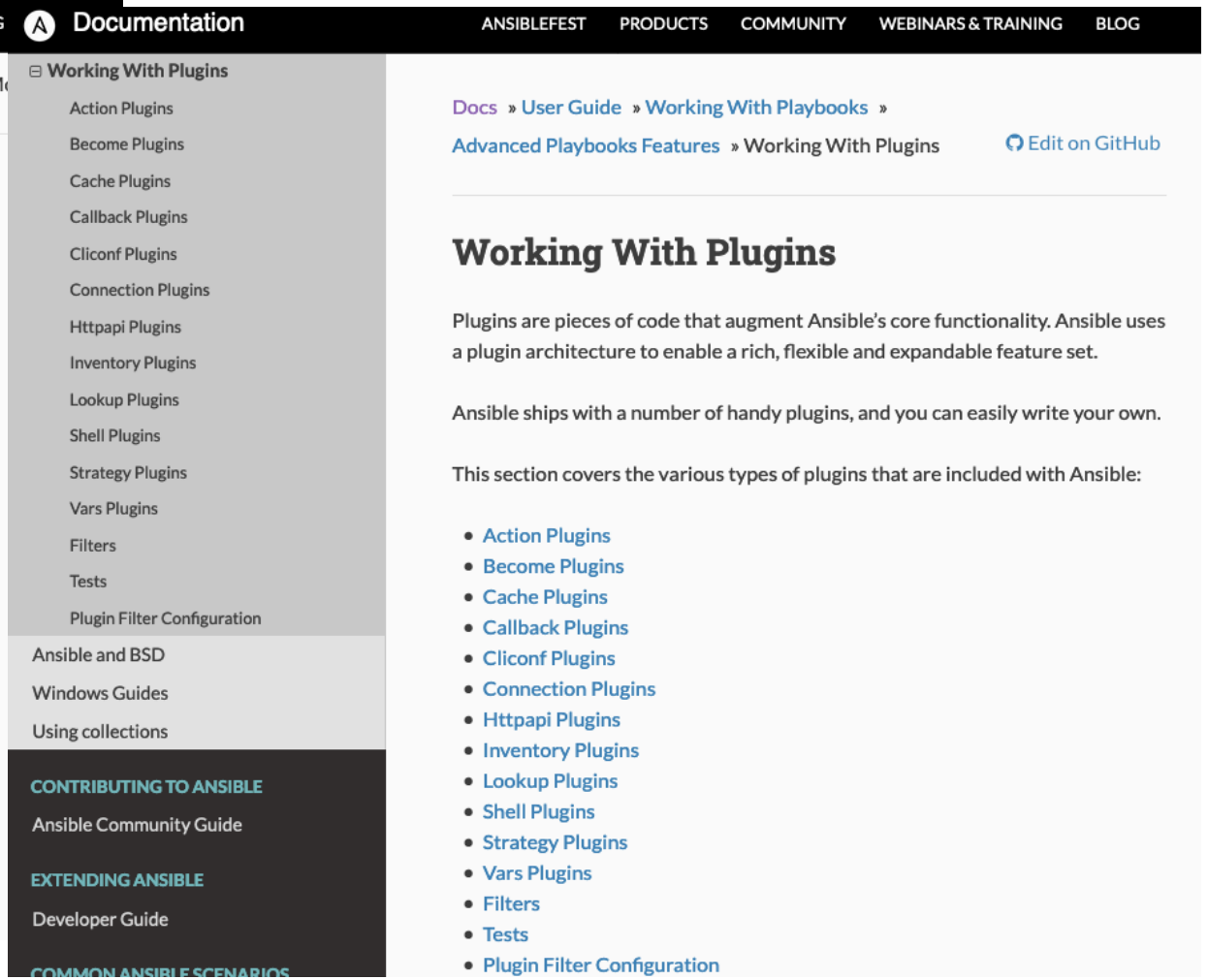
- **`Playbook`** is a YAML file which consists in a list of Plays.
  - A **`Play`** in a playbook is a list of Tasks.
    - A **`Task`** in a play contains Modules and its arguments.
      - **`Modules`** are the ones that do the actual work.

Ansible Architecture

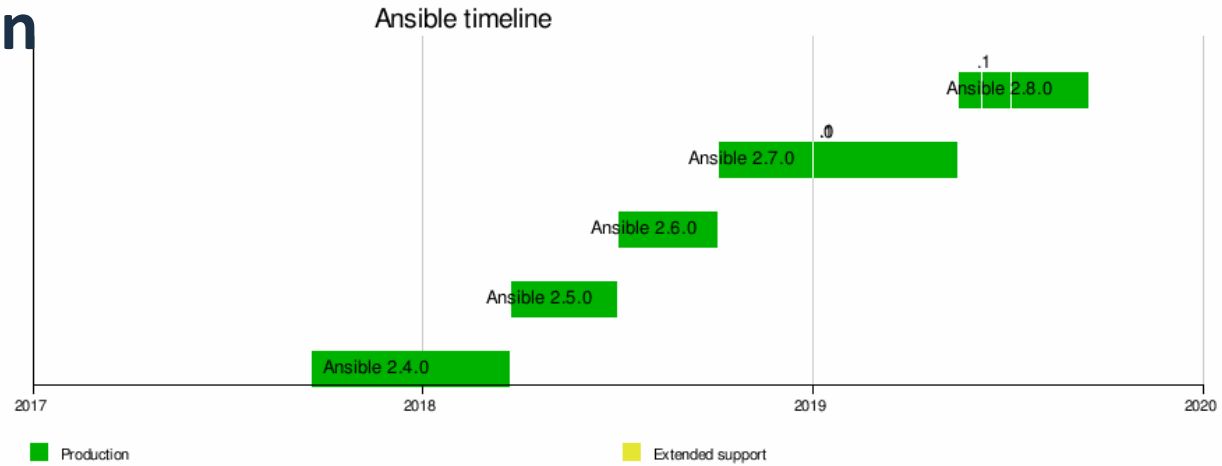


Modules & Plugins

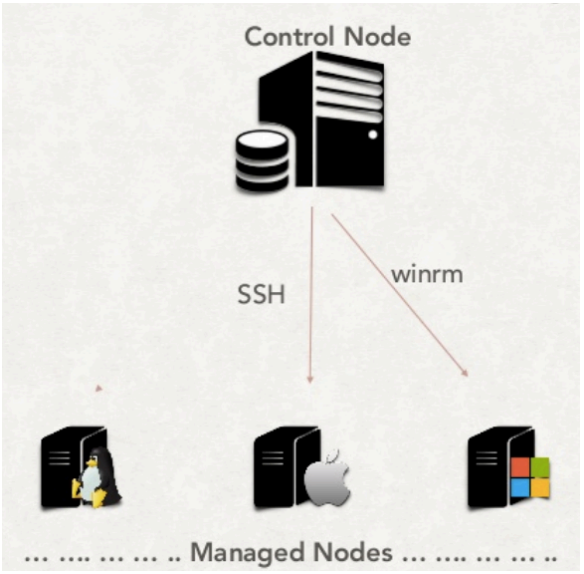




Installation



- Version:
  - ➢ latest
- Requirements:
  - ➢ ***Control Node***
    - **Python** 2 (v. 2.7) or Python 3 (v. 3.5 and higher)
    - Red Hat, Debian, CentOS, macOS, any of the BSDs, etc
      - ❖ **No Windows**
    - Nearness/closeness
  - ***Managed Nodes***
    - **Python** 2 (v. 2.7) or Python 3 (v. 3.5 and higher)
    - a way to communicate => **ssh**



Installation (2)

```
# Most common and preferred way of installation, from PyPI
$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
$ python get-pip.py [--user]
$ pip install [--user] ansible

# Ubuntu, needs PPA repo configured
# sudo apt install software-properties-common
$ sudo apt-add-repository -y ppa:ansible/ansible
$ sudo apt-get update
$ sudo apt-get install -y ansible

# CentOS, RHEl, S.L, needs epel-release rpm
# or use https://releases.ansible.com/ansible/rpm
$ sudo yum install -y ansible
```

Version, config files, demo ….

- On Mac

```
# MAC OsX
$ ansible --version
ansible 2.9.1
  config file = None
  configured module search path = [u'/Users/cristina/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /Users/cristina/Library/Python/2.7/lib/python/site-packages/ansible
  executable location = /Users/cristina/Library/Python/2.7/bin/ansible
  python version = 2.7.10 (default, Feb 22 2019, 21:55:15) [GCC 4.2.1 Compatible Apple LLVM 10.0.1 (clang-1001.0.37.14)]
```

Version, config files, demo ….

- On your hosts

```
# Ubuntu
dodas-ui:~$ ansible --version
ansible 2.6.20
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/cristina/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.12 (default, Oct  8 2019, 14:14:10) [GCC 5.4.0 20160609]

#CentOS

[root@form01b ~]$  ansible --version
ansible 2.6.20
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.5 (default, Aug  7 2019, 00:51:29) [GCC 4.8.5 20150623 (Red Hat 4.8.5-39)]
```

Config files

- $ANSIBLE_CONFIG
- {$PWD}/ansible.cfg
- ~/.ansible.cfg
- /etc/ansible/ansible.cfg

Config files

- $ANSIBLE_CONFIG
- {$PWD}/ansible.cfg
- ~/.ansible.cfg
- /etc/ansible/ansible.cfg

```
# Example config file for ansible -- https://ansible.com/
# ===================================================

# Nearly all parameters can be overridden in ansible-playbook
# or with command line flags. Ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory, or /etc/ansible/ansible.cfg, whichever it
# finds first

# For a full list of available options, run ansible-config list or see the
# documentation: https://docs.ansible.com/ansible/latest/reference_appendices/config.html.

[defaults]
inventory       = /etc/ansible/hosts
#library         = ~/.ansible/plugins/modules:/usr/share/ansible/plugins/modules
#module_utils    = ~/.ansible/plugins/module_utils:/usr/share/ansible/plugins/module_utils
#remote_tmp      = ~/.ansible/tmp
#local_tmp       = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks           = 5
#poll_interval   = 15
#sudo_user       = root
#ask_sudo_pass   = True
#ask_pass        = Tre
#transport       = smart
#remote_port    = 22
#module_lang    = C
#module_set_locale = False
```

Config files

- $ANSIBLE_CONFIG
- {$PWD}/ansible.cfg
- ~/.ansible.cfg
- /etc/ansible/ansible.cfg

```
# Example config file for ansible -- https://ansible.com/
# ================================================

# Nearly all parameters can be overridden in ansible-playbook
# or with command line flags. Ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory, or /etc/ansible/ansible.cfg, whichever it
# finds first

# For a full list of available options, run ansible-config list or see the
# documentation: https://docs.ansible.com/ansible/latest/reference_appendices/config.html.

[defaults]
inventory      = /etc/ansible/hosts
#library        = ~/.ansible/plugins/modules:/usr/share/ansible/plugins/modules
#module_utils   = ~/.ansible/plugins/module_utils:/usr/share/ansible/plugins/module_utils
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#module_lang    = C
#module_set_locale = False
```

ansible.cfg

```
1  [defaults]
2  host_key_checking = False
3  [galaxy]
4  #GALAXY_IGNORE_CERTS = True
```

Config files

- $ANSIBLE_CONFIG
- {$PWD}/ansible.cfg
- ~/.ansible.cfg
- /etc/ansible/ansible.cfg

```
$ ansible-config list
```

```
# Example config file for ansible -- https://ansible.com/
# ================================================

# Nearly all parameters can be overridden in ansible-playbook
# or with command line flags. Ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory, or /etc/ansible/ansible.cfg, whichever it
# finds first

# For a full list of available options, run ansible-config list or see the
# documentation: https://docs.ansible.com/ansible/latest/reference_appendices/config.html.

[defaults]
inventory      = /etc/ansible/hosts
#library        = ~/.ansible/plugins/modules:/usr/share/ansible/plugins/modules
#module_utils   = ~/.ansible/plugins/module_utils:/usr/share/ansible/plugins/module_utils
```

```
HOST_KEY_CHECKING:
  default: true
  description: Set this to "False" if you want to avoid host key checking by the underlying
    tools Ansible uses to connect to the host
  env:
  - {name: ANSIBLE_HOST_KEY_CHECKING}
  ini:
  - {key: host_key_checking, section: defaults}
  name: Check host keys
  type: boolean
```

EOSC-hub

CLI

- ansible - Define and run a single task ‘playbook’ against a set of hosts
- ansible-config - View ansible configuration
- ansible-console - REPL console for executing Ansible tasks
- ansible-doc - Plugin documentation tool
- ansible-galaxy - Perform various Role and Collection related operations
- ansible-inventory - Display or dump the configured inventory as Ansible sees it
- ansible-playbook-Runs Ansible playbooks, executing the defined tasks on the targeted hosts.
- ansible-pull - pulls playbooks from a VCS repo and executes them for the local host
- ansible-vault - encryption/decryption utility for Ansible data files

Inventory: formats, hosts, groups

- ***Formats***:
  - INI
  - YAML
- ***Hosts***
  - Remote nodes managed by Ansible
  - Can have individual **variables** (host name, service port number, etc, see ex…)
  - **Ranges:**
    - www[01:50].example.com
    - db-[a:f].example.com
  - **Vars:**

```
[group1]
host1 http_port=80 maxRequestsPerChild=808
host2 http_port=303 maxRequestsPerChild=909
```

```
[control]
controller ansible_host=localhost

[webservers]
web1 ansible_host=90.147.170.153

[dbservers]
db1 ansible_host=90.147.170.148

[lbservers]
lb1 ansible_host=90.147.170.151
```

```
all:
  hosts:
    mail.example.com:
  children:
    webservers:
      hosts:
        foo.example.com:
        bar.example.com:
    dbservers:
      hosts:
        one.example.com:
        two.example.com:
        three.example.com:
```

- ***Groups***
  - Used to clasify hosts, hosts in multiple groups
    - **what** hosts you are controlling at **what** times and for **what** purpose.
  - Default groups:
    - ***«all»***, ***«ungrouped»***

Inventory: formats, hosts, groups, vars

```
[atlanta]
host1
host2

[atlanta:vars]
ntp_server=ntp.atlanta.example.com
proxy=proxy.atlanta.example.com
```

- Vars:
  - Host vars
  - Group vars
    - Assigning a variable to **many** machines (see hosts_2.yaml)
    - Ansible **flattens** vars at level of host
      - internal rules for merging => order/precedence:
        - all group
        - parent group
        - child group
        - Host
      - When multiple inventory => order of the inventory source

```
a_group:
    testvar: a
    ansible_group_priority: 10
b_group:
    testvar: b
```

```
ansible-playbook get_logs.yml -i staging -i production
```

Modules & Run Commands

- ***Modules*** = units of code executed by Ansible
  = **«Ansible toolbox»**
  - Written in Python
  - Extensive library:
    - Web module index
    - `# ansible-doc –l`
  - ***(run)-commands => Ad-hoc commands***
    - ***command***
      - Exec commands on targets
    - ***shell***
      - Exec shell commands on targets
    - ***script***
      - Runs a local script on a remote node after transferring it
    - ***raw***
      - Exec a command without going throughthe Ansible module subsystem

```
dodas-ui:~$ ansible-doc [command|script|shell|raw]
```

- Cloud modules
- Clustering modules
- Commands modules
- Crypto modules
- Database modules
- Files modules
- Identity modules
- Inventory modules
- Messaging modules
- Monitoring modules
- Net Tools modules
- Network modules
- Notification modules
- Packaging modules
- Remote Management modules
- Source Control modules
- Storage modules
- System modules
- Utilities modules
- Web Infrastructure modules
- Windows modules

Ad-hoc Commands & Discovered Facts

```
cristina@dodas-ui:~$ cat hosts
[webservers]
90.147.170.44

[dbservers]
90.147.170.49

[lbservers]
90.147.170.42
```

```
cristina@dodas-ui:~$ ansible all -i hosts -u ubuntu -m ping
90.147.170.44 | FAILED! => {
    "changed": false,
    "module_stderr": "Shared connection to 90.147.170.44 closed.\r\n",
    "module_stdout": "/bin/sh: 1: /usr/bin/python: not found\r\n",
    "msg": "MODULE FAILURE",
    "rc": 127
}
90.147.170.49 | FAILED! => {
    "changed": false,
    "module_stderr": "Shared connection to 90.147.170.49 closed.\r\n",
    "module_stdout": "/bin/sh: 1: /usr/bin/python: not found\r\n",
    "msg": "MODULE FAILURE",
    "rc": 127
}
90.147.170.42 | FAILED! => {
    "changed": false,
    "module_stderr": "Shared connection to 9
    "module_stdout": "/bin/sh: 1: /usr/bin/p
    "msg": "MODULE FAILURE",
    "rc": 127
}
```

```
cristina@dodas-ui:~$ ansible all -i hosts -u ubuntu -m ping
90.147.170.49 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
90.147.170.42 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
90.147.170.44 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Ad-hoc Commands & Discovered Facts (2)

```
cristina@dodas-ui:~$ ansible all -i hosts -u ubuntu -m setup
90.147.170.49 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses":
            "90.147.170.49"
        ],
        "ansible_all_ipv6_addresses":
            "fe80::f816:3eff:fe61:50ec
        ],
        "ansible_apparmor": {
            "status": "enabled"
        },
        "ansible_architecture": "x86_6
        "ansible_bios_date": "04/01/20
        "ansible_bios_version": "Ubunt
        "ansible_cmdline": {
            "BOOT_IMAGE": "/boot/vmlir
            "console": "ttyS0",
            "ro": true,
            "root": "LABEL=cloudimg-rc
        },
```

```
90.147.170.44 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses"
            "90.147.170.44"
        ],
        "ansible_all_ipv6_addresses"
            "fe80::f816:3eff:fed2:97
        ],
        "ansible_apparmor": {
            "status": "enabled"
        },
        "ansible_architecture": "x86_
        "ansible_bios_date": "04/01/
        "ansible_bios_version": "Ubu
        "ansible_cmdline": {
            "BOOT_IMAGE": "/boot/vml
            "console": "ttyS0",
            "ro": true,
            "root": "LABEL=cloudimg-
        },
```

```
90.147.170.42 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "90.147.170.42"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::f816:3eff:feef:7052"
        ],
        "ansible_apparmor": {
            "status": "enabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "04/01/2014",
        "ansible_bios_version": "Ubuntu-1.8.2-1ubuntu1",
        "ansible_cmdline": {
            "BOOT_IMAGE": "/boot/vmlinuz-4.4.0-21-generic",
            "console": "ttyS0",
            "ro": true,
            "root": "LABEL=cloudimg-rootfs"
        },
```

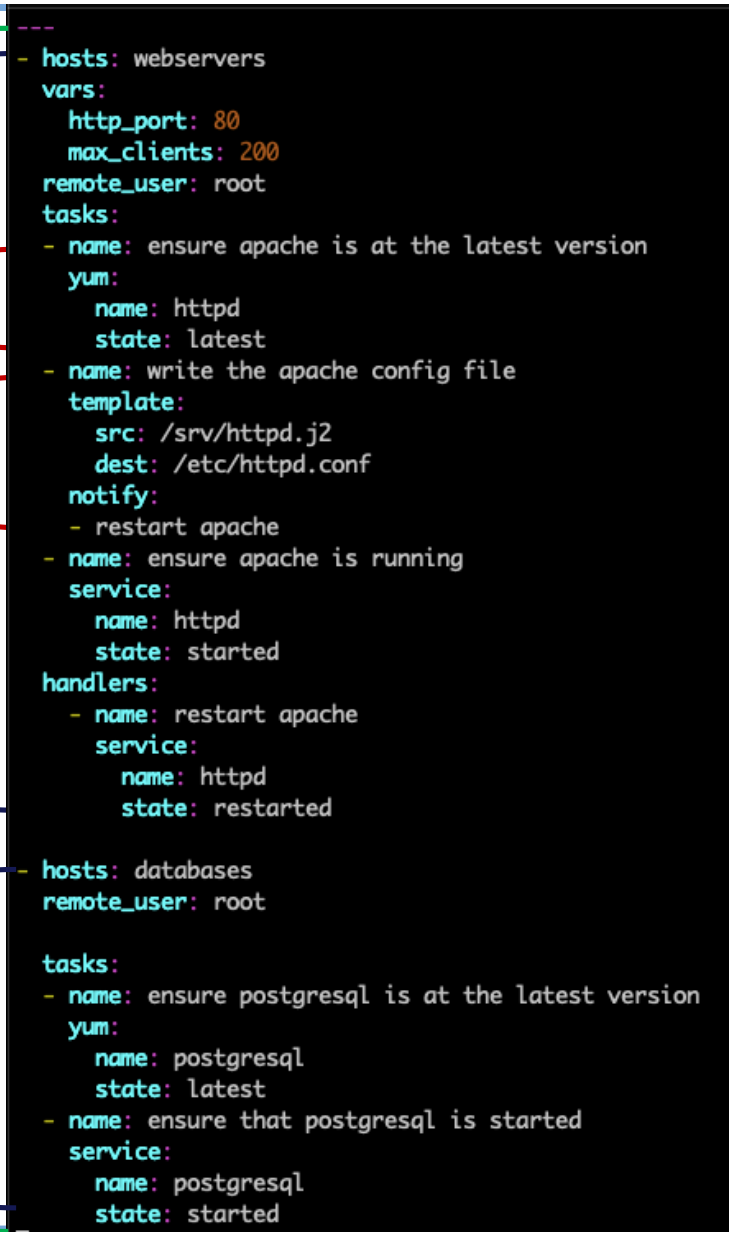
Ad-hoc Commands & Discovered Facts (3)

```
cristina@dodas-ui:~$ ansible webservers -i hosts -u ubuntu -m apt -a "name=apache2 state=present" -b
90.147.170.44 | SUCCESS => {
    "cache_update_time": 1573949351,
    "cache_updated": false,
    "changed": true,
    "stderr": "",
    "stderr_lines": [],
    "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe following additional packages will be installed:\n
```

```
cristina@dodas-ui:~$ ansible webservers -i hosts -u ubuntu -m apt -a "name=apache2 state=present" -b
90.147.170.44 | SUCCESS => {
    "cache_update_time": 1573949351,
    "cache_updated": false,
    "changed": false
}
```

Playbooks, plays, tasks

- ***Task***
  - Application of a single module on one or more hosts
  - Each task ends in a well-defined state
- ***Play***
  - A set of ordered tasks, associated with a group of hosts
- ***Playbook***
  - Associate the hosts with the desired state of the infrastructure, defining the set of tasks to be performed
  - They therefore allow orchestration and deployment
  - Collection of plays



```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running
    service:
      name: httpd
      state: started
  handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted

- hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is at the latest version
    yum:
      name: postgresql
      state: latest
  - name: ensure that postgresql is started
    service:
      name: postgresql
      state: started
```

Creating Reusable Playbooks - Roles

- Including (A. 2.0) and Importing (A. 2.4)
  - Dynamic vs static
- Roles
  - decompose **complex jobs** into **smaller** pieces
    - organizing **multiple**, **related** Tasks and **encapsulating data** needed to accomplish those Tasks
      - Variables, handlers, modules, plugins
  - **special kind of Playbooks**, fully **self-contained**, with tasks, variables, configuration templates, other supporting files
    - **cannot** be executed
  - provide a **skeleton** for an **independent** and **reusable collection** of variables, tasks, templates, files, and modules which can be **automatically loaded** into the playbook.
    - Playbooks are a **collection of roles**
    - Every role has **specific functionality**

Roles - Location

- ***Location***:
  - Search path
    - A ***roles***/ directory, relative to the playbook file.
    - By default, in ***/etc/ansible/roles***
  - Defined in the configuration, can be customized

```
dodas-ui:~$ ansible-config dump| grep -i roles
DEFAULT_ROLES_PATH(default) = [u'/home/cristina/.ansible/roles', u'/usr/share/ansible/roles',
u'/etc/ansible/roles']
```

```
[root@form01b ~]$  ansible-config dump |grep -i roles
DEFAULT_ROLES_PATH(default) = ['/root/.ansible/roles', '/usr/share/ansible/roles', '/etc/ansible/roles']
```

  - Best-practice => define it (*ansible.cfg*) in a ***«project»*** related directory

```
[defaults]
roles_path = ~/ansible_project/roles
```

Roles - Directory Structure

- Expect files to be in certain **directory names**
  - **At least one** of the listed directories
  - When exists – mut contain «***main.yml***»
- Content:
  - ***tasks*** - main list of tasks to be executed by the role.
  - ***handlers*** - handlers, which may be used by this role or even anywhere outside this role.
  - ***defaults*** - default variables for the role (see Using Variables for more information).
  - ***vars*** - other variables for the role
  - ***files*** - contains files which can be deployed via this role
  - ***templates*** - templates which can be deployed
  - ***meta*** - defines some meta data for this role.

```
roles/
`-- rolename
    |-- defaults
    |   `-- main.yml
    |-- files
    |-- handlers
    |   `-- main.yml
    |-- meta
    |   `-- main.yml
    |-- README.md
    |-- tasks
    |   `-- main.yml
    |-- templates
    |-- tests
    |   |-- inventory
    |   `-- test.yml
    `-- vars
        `-- main.yml
```

Roles – how to use

- Classic/original - via the *roles:* option for a given play

```
---
- hosts: webservers
  roles:
    - common
    - webservers
```

- Order to **add in the play/playbook**:
  - roles/x/***tasks/main.yml***
  - roles/x/***handlers/main.yml***
  - roles/x/***vars/main.yml***
  - roles/x/***meta/main.yml***
  - Any copy, script, template can reference files in roles/x***/{files,templates,tasks}/***

- Order of **execution** of the playbook
  - ***Each role*** listed in *roles*
    - ***Any role dependencies*** defined in the *meta/main.yml*
  - ***Any tasks*** defined in the play.
  - ***Any handlers*** triggered so far will be run.

From monolithic playbook to roles

```yaml
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: write the index.html file
    template:
      src: /src/index.html.j2
      dest: /var/www/html
  - name: copy httpd.conf
    copy: src=/srv/httpd.conf dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running, enabled at boot
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted
```

```
ansible_project/
|-- ansible.cfg
|-- ansible_playbook.yml
|-- hosts
|-- roles
|   `-- apache_role
```

```
apache_role/
|-- defaults
|   `-- main.yml
|-- files
|-- handlers
|   `-- main.yml
|-- meta
|   `-- main.yml
|-- README.md
|-- tasks
|   `-- main.yml
|-- templates
|-- tests
|   |-- inventory
|   `-- test.yml
`-- vars
    `-- main.yml
```

Extracting Tasks

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: write the index.html file
    template:
      src: /src/index.html.j2
      dest: /var/www/html
  - name: copy httpd.conf
    copy: src=/srv/httpd.conf dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running, enabled at boot
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted
```

```
dodas-ui:~/ansible_project$ cat roles/apache_role/tasks/install.yml
---
- name: ensure apache is at the latest version
  yum: name=httpd state=latest
```

```
dodas-ui:~/ansible_project$ cat roles/apache_role/tasks/configure.yml
---
- name: write the apache config file
  template:
    src: templates/index.html.j2
    dest: /var/www/html
- name: copy httpd.conf
  copy: src=files/httpd.conf dest=/etc/httpd.conf
  notify:
  - restart apache
```

```
dodas-ui:~/ansible_project$ cat roles/apache_role/tasks/service.yml
---
- name: ensure apache is running, enabled at boot
  service: name=httpd state=started enabled=yes
```

```
dodas-ui:~/ansible_project$ cat roles/apache_role/tasks/main.yml
---
# tasks file for apache
- include: install.yml
- include: configure.yml
- include: service.yml
```

Extracting handler

```yaml
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: write the index.html file
    template:
      src: /src/index.html.j2
      dest: /var/www/html
  - name: copy httpd.conf
    copy: src=/srv/httpd.conf dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running, enabled at boot
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted
```

```
dodas-ui:~/ansible_project$ cat roles/apache_role/handlers/main.yml
---
# handlers file for apache
- name: restart apache
  service:
    name: httpd
    state: restarted
```

Variables

- two types of variables that can be defined in a role:
  - ***role variables***, loaded from *roles/<role_name>/vars/main.yaml*
    - used for example for system-specific constants that don't change much
  - ***role defaults***, which are loaded from *roles/<role_name>/defaults/main.yaml*
    - place holders for actual data, a reference of what variables a developer may be interested in defining with site-specific values
- Main difference – **precedence** order
  - *Defaults* - are the lowest order variables

Variables in roles - examples

```
dodas-ui:~/ansible_project$ cat roles/apache_role/vars/main.yml
---
# vars file for apache - RedHat specific
apache_service: httpd
apache_daemon: httpd
apache_daemon_path: /usr/sbin/
apache_server_root: /etc/httpd
apache_conf_path: /etc/httpd/conf.d
```

```
dodas-ui:~/ansible_project$ cat roles/apache_role/defaults/main.yml
---
# defaults file for apache
apache_test_message: This is a test message
```

```
dodas-ui:~/ansible_project$ cat roles/apache_role/templates/index.html.j2
{{ apache_test_message }} {{ ansible_distribution }} {{ ansible_distribution_version }}  <br>
Current Host: {{ ansible_hostname }} <br>
Server list: <br>
{% for host in groups.webservers %}
{{ host }} <br>
{% endfor %}
```

```
dodas-ui:~/ansible_project$ ansible localhost -m setup 2>/dev/null |grep -w 'ansible_distribution
\|ansible_distribution_version\|ansible_hostname'
        "ansible_distribution": "Ubuntu",
        "ansible_distribution_version": "16.04",
        "ansible_hostname": "dodas-ui",
```

Moving config files

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: write the index.html file
    template:
      src: /src/index.html.j2
      dest: /var/www/html
  - name: copy httpd.conf
    copy: src=/srv/httpd.conf dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running, enabled at boot
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted
```

```
dodas-ui:~/ansible_project$ tree roles/apache_role/templates/
roles/apache_role/templates/
`-- index.html.j2
```

```
dodas-ui:~/ansible_project$ tree roles/apache_role/files/
roles/apache_role/files/
`-- httpd.conf
```

New playbook that uses the new role

```
dodas-ui:~/ansible_project$ cat site.yml
---
- hosts: webservers
  roles:
  - apache_role
```

Check

```
dodas-ui:~/ansible_project$ ansible webservers -i hosts -u centos -m ping
90.147.170.153 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Check

```
dodas-ui:~/ansible_project$ ansible-lint site.yml
[403] Package installs should not use latest
/home/cristina/ansible_project/roles/apache_role/tasks/install.yml:2
Task/Handler: ensure apache is at the latest version
```

PLAY

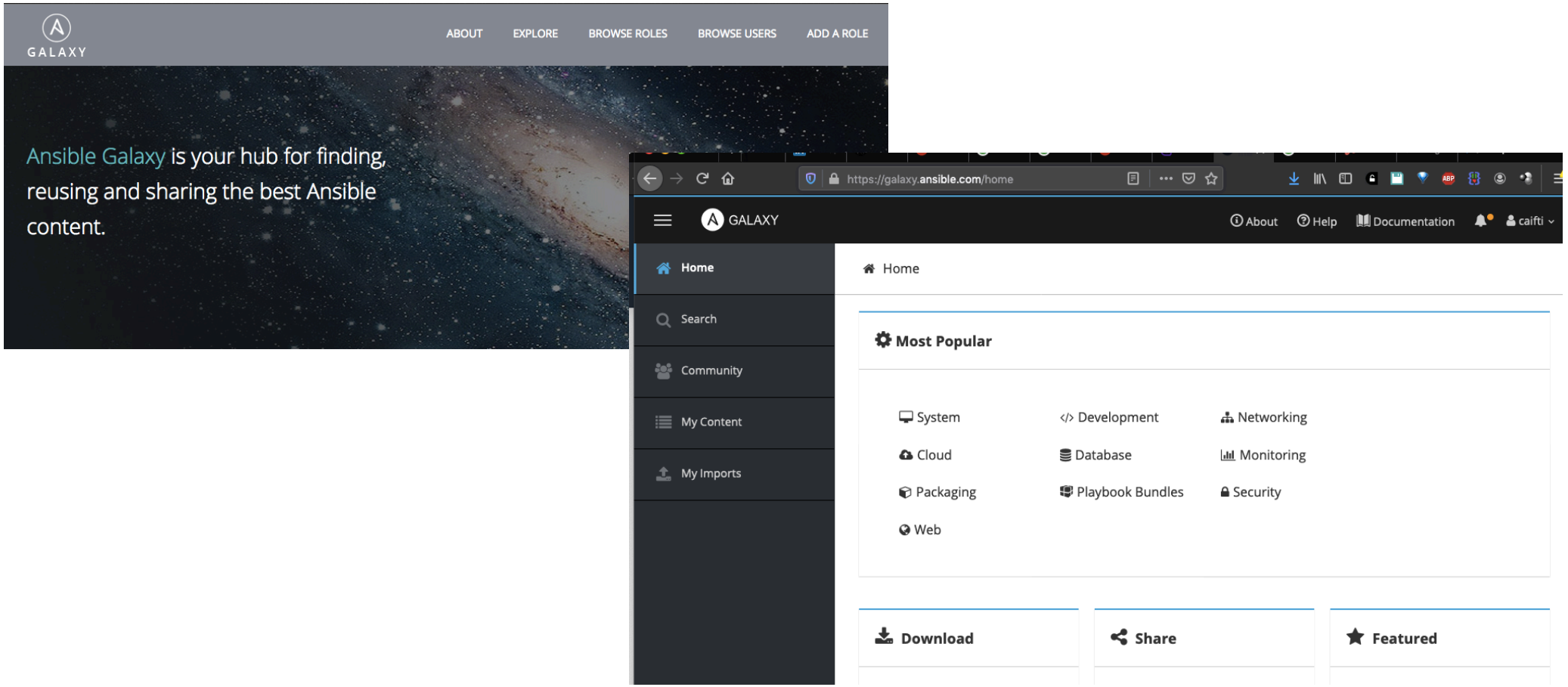
Reference: https://docs.ansible.com/ansible-lint/rules/default_rules.html

New playbook that uses the new role

```
dodas-ui:~/ansible_project$ ansible-playbook -i hosts -u centos site.yml

PLAY [webservers] ***************************************************************

TASK [Gathering Facts] **********************************************************
ok: [90.147.170.153]

TASK [apache_role : ensure apache is at the latest version] *********************
changed: [90.147.170.153]

TASK [apache_role : write the apache config file] *******************************
changed: [90.147.170.153]

TASK [apache_role : copy httpd.conf] ********************************************
changed: [90.147.170.153]

TASK [apache_role : ensure apache is running, enabled at boot] ******************
changed: [90.147.170.153]

RUNNING HANDLER [apache_role : restart apache] **********************************
changed: [90.147.170.153]

PLAY RECAP **********************************************************************
90.147.170.153             : ok=6    changed=5    unreachable=0    failed=0
```

PLAY

Ansible Galaxy – Reusing Roles

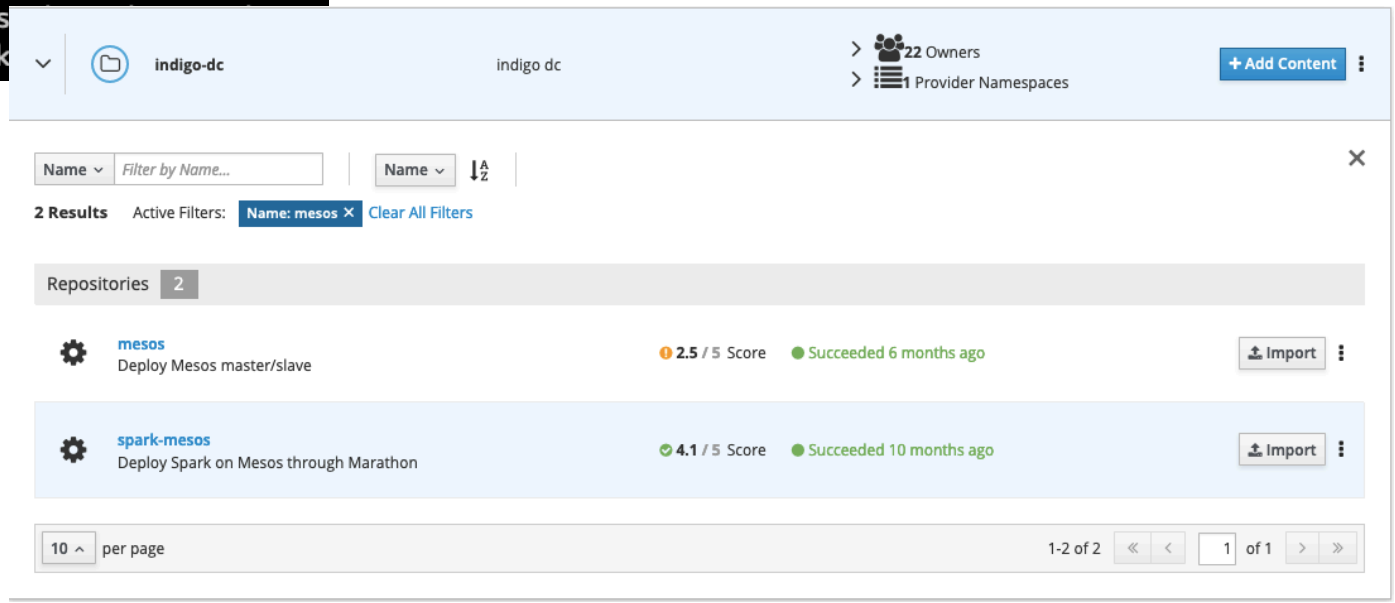


ansible-galaxy CLI tool

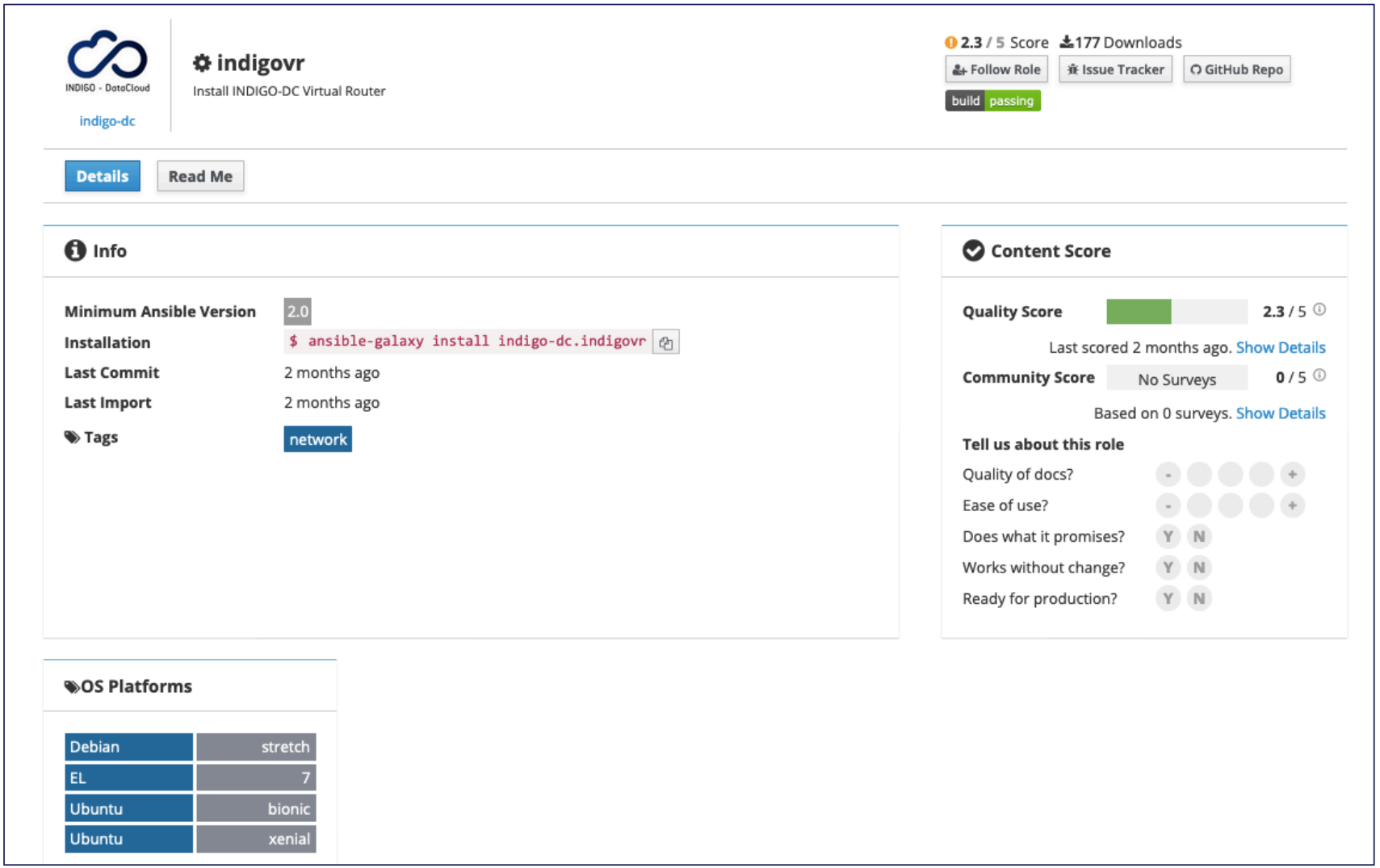
```
dodas-ui:~$ ansible-galaxy --help
Usage: ansible-galaxy [delete|import|info|init|install|list|login|remove|search|setup] [--help] [options] ...

Options:
  -h, --help            show this help message and exit
  -c, --ignore-certs    Ignore SSL certificate validation errors.
  -s API_SERVER, --server=API_SERVER
                        The API server destination
  -v, --verbose         verbose mode (-vvv for more, -vvvv to enable
                        connection debugging)
  --version             show program's version number and exit

 See 'ansible-galaxy <command> --help' for more information on a specific
command.
```

Search Roles

```
dodas-ui:~$ ansible-galaxy search mesos --author indigo-dc

Found 7 roles matching your search:

 Name                  Description
 ----                  -----------
 indigo-dc.calico      Configure and run calico
 indigo-dc.chronos     Deploy Chronos Framework
 indigo-dc.consul      Deploy consul agent/server with dnsmasq
 indigo-dc.marathon    Deploy Marathon
 indigo-dc.mesos       Deploy Mesos master/slave
 indigo-dc.spark-mesos Deploy Spark on Mesos
 indigo-dc.zookeeper   Deploy zookeeper dock
```



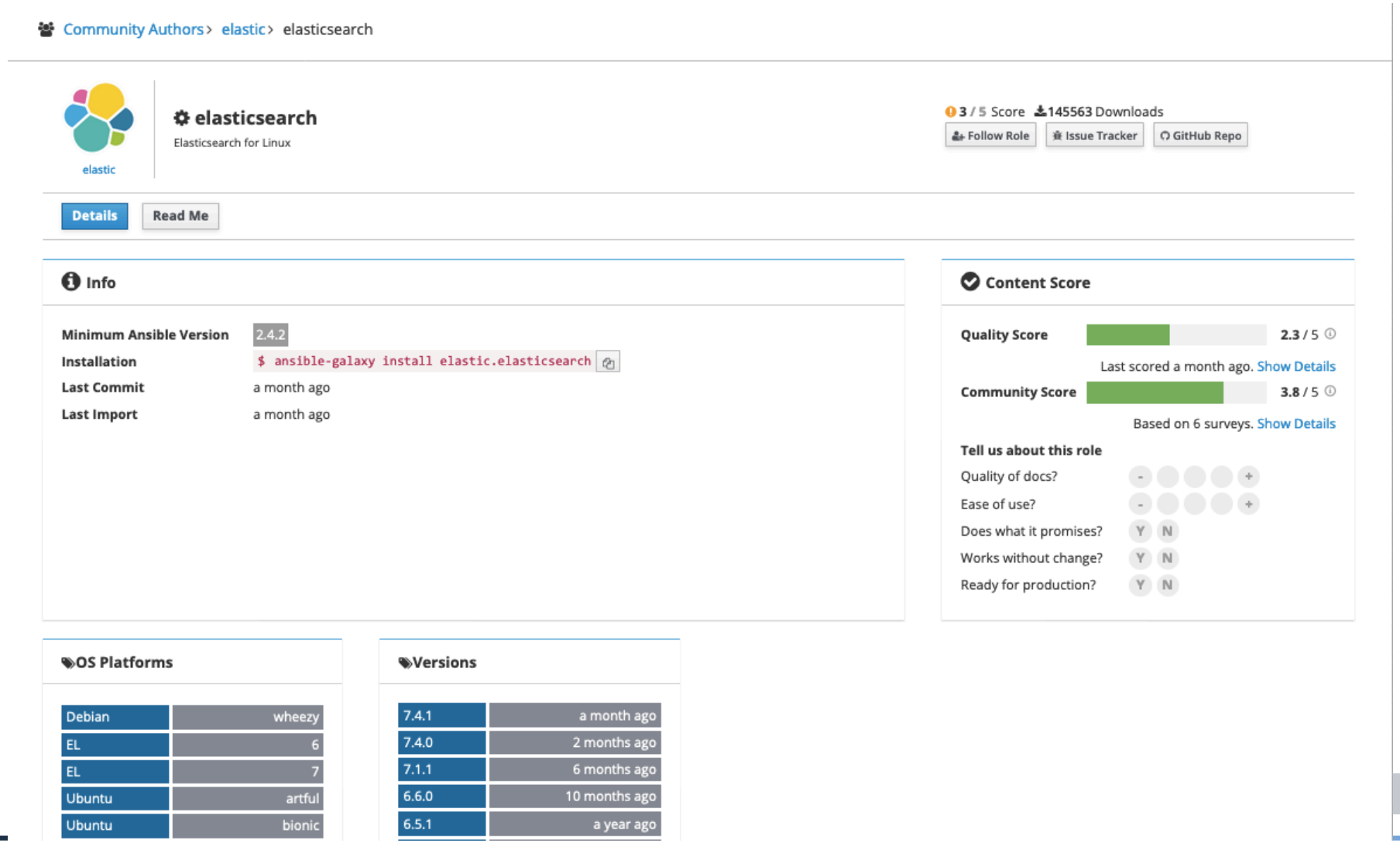
Get Info


INDIGO - DataCloud
indigo-dc
indigovr
Install INDIGO-DC Virtual Router
2.3 / 5 Score
177 Downloads
Follow Role
Issue Tracker
GitHub Repo
build passing
Details
Read Me
Info
Minimum Ansible Version
2.0
Installation
$ ansible-galaxy install indigo-dc.indigovr
Last Commit
2 months ago
Last Import
2 months ago
Tags
network
Content Score
Quality Score
2.3 / 5
Last scored 2 months ago. Show Details
Community Score
No Surveys
0 / 5
Based on 0 surveys. Show Details
Tell us about this role
Quality of docs?
Ease of use?
Does what it promises?
Works without change?
Ready for production?
OS Platforms
Debian stretch
EL 7
Ubuntu bionic
Ubuntu xenial

Get Info (2)

```
dodas-ui:~$ ansible-galaxy info indigo-dc.indigovr

Role: indigo-dc.indigovr
        description: Install INDIGO-DC Virtual Router
        active: True
        commit: a19811b095621245265da1992f85df109f878151
        commit_message: Re-use recipes for certificate signing in vrouter and standalone
        commit_url: https://api.github.com/repos/indigo-dc/ansible-role-indigovr/git/commits/a19811b0956
        company: INDIGO-DataCloud
        created: 2018-06-21T06:52:35.537603Z
        download_count: 177
        forks_count: 4
        github_branch: master
        github_repo: ansible-role-indigovr
        github_user: indigo-dc
        id: 26429
        imported: 2019-10-08T03:21:16.554491-04:00
        is_valid: True
        issue_tracker_url: https://github.com/indigo-dc/ansible-role-indigovr/issues
        license: Apache
        min_ansible_version: 2.0
        modified: 2019-10-08T07:21:16.563710Z
        open_issues_count: 1
        path: [u'/home/cristina/.ansible/roles', u'/usr/share/ansible/roles', u'/etc/ansible/roles']
        role_type: ANS
        stargazers_count: 2
        travis_status_url: https://travis-ci.org/indigo-dc/ansible-role-indigovr.svg?branch=master
```

One more!!


Community Authors› elastic› elasticsearch
elastic
elasticsearch
Elasticsearch for Linux
3 / 5 Score
145563 Downloads
Follow Role
Issue Tracker
GitHub Repo
Details
Read Me
Info
Minimum Ansible Version
2.4.2
Installation
$ ansible-galaxy install elastic.elasticsearch
Last Commit
a month ago
Last Import
a month ago
Content Score
Quality Score
2.3 / 5
Last scored a month ago. Show Details
Community Score
3.8 / 5
Based on 6 surveys. Show Details
Tell us about this role
Quality of docs?
Ease of use?
Does what it promises?
Works without change?
Ready for production?
OS Platforms
Debian wheezy
EL 6
EL 7
Ubuntu artful
Ubuntu bionic
Versions
7.4.1 a month ago
7.4.0 2 months ago
7.1.1 6 months ago
6.6.0 10 months ago
6.5.1 a year ago

Download and Install – from Galaxy

```
ansible-galaxy install role_name(s)[,version]
```

Where:

```
[root@form08 ansible_project]# ansible-config dump |grep -i roles
DEFAULT_ROLES_PATH(/root/ansible_project/ansible.cfg) = [u'/root/ansible_project/roles']
```

Do:

```
[root@form08 ansible_project]# ansible-galaxy install elastic.elasticsearch,7.4.1
- downloading role 'elasticsearch', owned by elastic
- downloading role from https://github.com/elastic/ansible-elasticsearch/archive/7.4.1.tar.gz
- extracting elastic.elasticsearch to /root/ansible_project/roles/elastic.elasticsearch
- elastic.elasticsearch (7.4.1) was installed successfully
```

Check:

```
[root@form08 ansible_project]# ansible-galaxy list
- elastic.elasticsearch, 7.4.1
```

Download and Install – from Github

```
ansible-galaxy install scm+role_repo_url[,version]
```

```
[root@form08 ansible_project]# ansible-galaxy install git+https://github.com/elastic/ansible-elasticsearch.git,7.4.1
- extracting ansible-elasticsearch to /root/ansible_project/roles/ansible-elasticsearch
- ansible-elasticsearch (7.4.1) was installed successfully
```

```
[root@form08 ansible_project]# ansible-galaxy list
- ansible-elasticsearch, 7.4.1
```

Creating roles with ansible-galaxy

- **ansible-galaxy** tool can also be used to generate ***scaffolding***, an initial set of files and directories involved in a role:

```
[ansible_project]# ansible-galaxy init apache_new
-   apache_new was created successfully

[ansible_project]# ansible-galaxy list
- ansible-elasticsearch, 7.4.1
- apache_new, (unknown version)

[ansible_project]# ansible-galaxy init --init-path=INIT_PATH apache_new
```

```
apache_new/
|-- defaults
|   `-- main.yml
|-- files
|-- handlers
|   `-- main.yml
|-- meta
|   `-- main.yml
|-- README.md
|-- tasks
|   `-- main.yml
|-- templates
|-- tests
|   |-- inventory
|   `-- test.yml
`-- vars
    `-- main.yml
```

Importing roles – using CLI & WebUI

- CLI
  - GitHub repository for new role
  - login to Ansible Galaxy
  - ansible import

```
# ansible-galaxy login –h
Usage: ansible-galaxy login [options]

Options:
  --github-token=TOKEN  Identify with github token rather than
username and
                        password.
  -h, --help            show this help message and exit
  -c, --ignore-certs    Ignore SSL certificate validation
errors.
  -s API_SERVER, --server=API_SERVER
                        The API server destination
  -v, --verbose         verbose mode (-vvv for more, -vvvv to
enable
                        connection debugging)
  --version             show program's version number and exit
```

ansible login

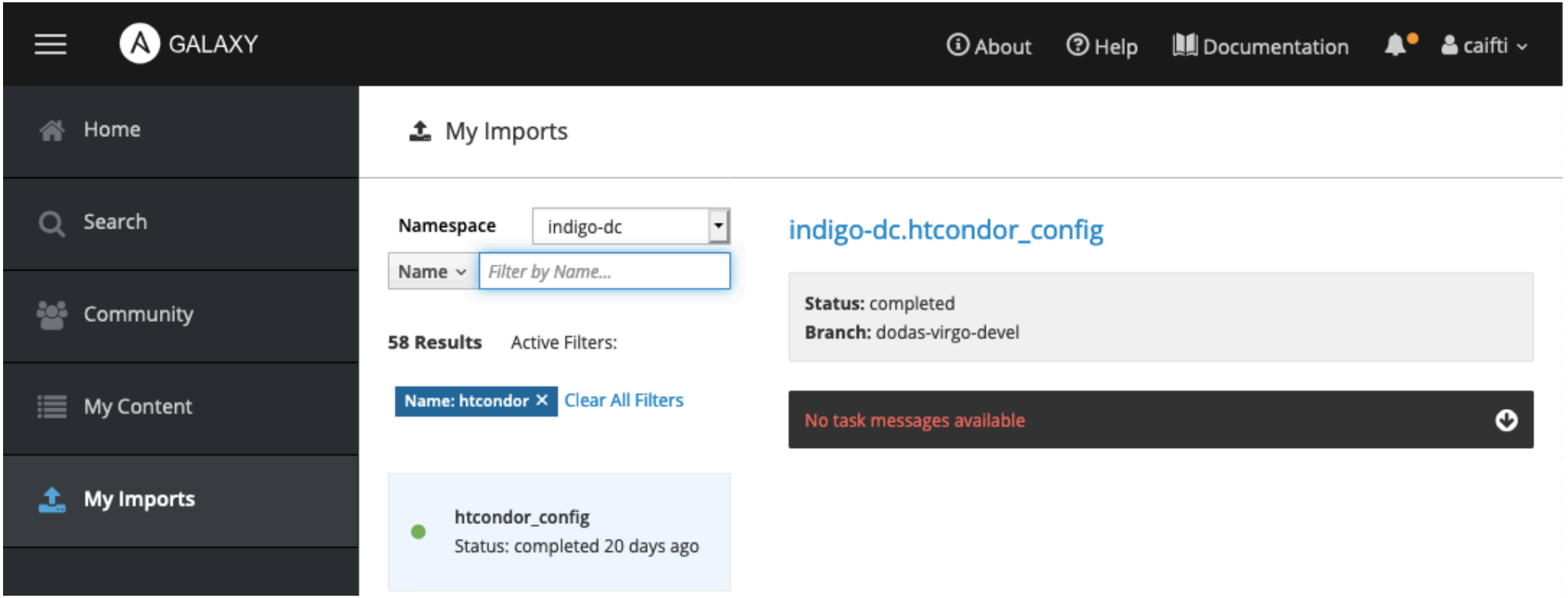
```
# ansible-galaxy login –h
Usage: ansible-galaxy login [options]

Options:
  --github-token=TOKEN  Identify with github token rather than username and
                        password.
  -h, --help            show this help message and exit
  -c, --ignore-certs    Ignore SSL certificate validation errors.
  -s API_SERVER, --server=API_SERVER
                        The API server destination
  -v, --verbose         verbose mode (-vvv for more, -vvvv to enable
                        connection debugging)
  --version             show program's version number and exit
```

Github token can be generated here: https://github.com/settings/tokens

ansible-galaxy import

```
# ansible-galaxy import -h
Usage: ansible-galaxy import [options] github_user github_repo

Options:
  --branch=REFERENCE    The name of a branch to import. Defaults to the epository's default
                         branch (usually master)
  -h, --help            show this help message and exit
  -c, --ignore-certs    Ignore SSL certificate validation errors.
  --no-wait             Don't wait for import results.
  --role-name=ROLE_NAME
                        The name the role should have, if different than the
                        repo name
  -s API_SERVER, --server=API_SERVER
                        The API server destination
  --status              Check the status of the most recent import request for
                        given github_user/github_repo.
  -v, --verbose         verbose mode (-vvv for more, -vvvv to enable
                        connection debugging)
  --version             show program's version number and exit
```

Import using Ansible Galaxy Web GUI



Ansible – advanced usage

- Debbuging
- Optmization
- Vault

Verbose & debug

- Verbose flag: ***-vvv*** or ***–verbose***
  - prints all the values that were returned by each module after it runs

```
# ansible-playbook --verbose playbook.yml
```

- ***debug*** module - prints statements during execution and can be useful for debugging variables or expressions without necessarily halting the playbook. Useful for debugging together with the 'when:' directive.

```
- debug: var=myvariable
- debug: msg="The value of myvariable is {{ var }}"
- debug:
    msg: "System {{ inventory_hostname }} has gateway {{ ansible_default_ipv4.gateway }}"
  when: ansible_default_ipv4.gateway is defined
```

assert & pause

- ***Assert module*** - module asserts that given expressions are true
  - `assert: { that: "ansible_os_family != 'RedHat'» }`

- ***Pause module*** - pauses playbook execution for a **set amount of time**, or until a prompt is acknowledged
  - default behavior is to **pause with a prompt**

```
# Pause for 5 minutes to build app cache.
- pause:
    minutes: 5
```

syntax check & list tasks

- *«--syntax-check»* perform a syntax check on the playbook, but do not execute it

```
dodas-ui:~$ cd ansible_project/
dodas-ui:~/ansible_project$ ansible-playbook --syntax-check -i hosts site.yml

playbook: site.yml
dodas-ui:~/ansible_project$ ansible-playbook --syntax-check -i hosts ansible_playbook.yml

playbook: ansible_playbook.yml
```

- *«--list-tasks»* list all tasks that would be executed

```
dodas-ui:~/ansible_project$ ansible-playbook --list-tasks -i hosts ansible_playbook.yml

playbook: ansible_playbook.yml

  play #1 (webservers): webservers      TAGS: []
    tasks:
      ensure apache is at the latest version    TAGS: []
      write the index.html file TAGS: []
      copy httpd.conf    TAGS: []
      ensure apache is running, enabled at boot TAGS: []
```

```
dodas-ui:~/ansible_project$ ansible-playbook --list-tasks -i hosts site.yml

playbook: site.yml

  play #1 (webservers): webservers      TAGS: []
    tasks:
      apache_role : ensure apache is at the latest version      TAGS: []
      apache_role : write the apache config file        TAGS: []
      apache_role : copy httpd.conf     TAGS: []
      apache_role : ensure apache is running, enabled at boot   TAGS: []
```

Optimization (1)

- ***SSH multiplexing & ControlPersist***
  - When Ansible runs a playbook, it will make **many SSH connections**, in order to do things such as copy over files and run commands.
  - Each time Ansible makes a new SSH connection to a host, it has to pay the **negotiation penalty**.
  - OpenSSH supports an optimization called ***SSH multiplexing***, which is also referred to as ***ControlPersist***:
    - a master connection is opened for each host and a control socket is used to communicate with the remote host instead of making a new TCP connection

- In Ansible:

```
ControlMaster default=auto
ControlPath default=$HOME/.ansible/cp/ansible-ss-%h-%p-%r
ControlPersist 60s
```

```
ANSIBLE_SSH_ARGS(default) = -C -o ControlMaster=auto -o ControlPersist=60s
```

Optimization (2)

- ***Pipelining***
  - When Ansible executes a task
    - It generates a Python script based on the module being invoked
    - Then it copies the Python script to the host
    - Finally, it executes the Python script
  - Enabling **pipelining** reduces the number of SSH operations required to execute a module on the remote server
    - by **executing many** ansible modules **without** actual **file transfer**.
    - this can result in a **very significant performance improvement** when enabled
    - however when using "***sudo***:" operations you must first ***disable 'requiretty'*** in /etc/sudoers on all managed hosts.

Optimization (3)

- Facts caching
  - When a fact cache is enabled and there is valid data for a host, Ansible will use that rather than running an implicit setup job on a remote host.
  - Plugins => ***# ansible-doc -t cache -l***

    - jsonfile JSON formatted files.
    - memcached Use memcached DB for cache
    - memory RAM backed, non persistent
    - mongodb Use MongoDB for caching
    - pickle Pickle formatted files.
    - redis Use Redis DB for cache
    - yaml YAML formatted files.

  - `[defaults]`
  - `gathering = smart fact`
  - `_caching_timeout = 86400`
    `fact_caching = ... .`

Example – using elastic.elasticsearch module

- Connect to your VM & become root

```
# ssh -i ~/ssh_keys/id_rsa_pg -l centos 193.204.89.119
```

- Get project from baltig

```
# git clone https://baltig.infn.it/aiftim/corso_dodas_2019.git
```

- Update files to meet your environment – ansible.cfg, hosts…
- Install *elasticsearch* role

```
# ansible-galaxy install elastic.elasticsearch,7.4.1
```

- Check ….
- Run

```
# ansible-playbook -i hosts es.yaml
```

Thank you for your attention!

Questions?

Contact

eosc-hub.eu @EOSC_eu