**PROMO DEVOPS 032019**

**ANSIBLE**

**Galaxy**

**Prérequis:**

Faire le TP setUp-Dev-Env

Faire le TP Playbooks

Faire le TP 3-tiers

**Besoin Client :**

Socle infrastructure afin d’héberger son application JAVA écrite en JAVA EE 8

L’application doit être livrée en production dans deux mois.

Les playbooks fournis pour déployer l’architecture choisie est bon début.

Il nous faut maintenant penser au futur de l’application, c’est à dire son cycle de vie dans le système d’information, et à son déploiement dans les différents environnements : DEVELOPPEMENT, INTEGRATION, PRE-PRODUCTION, PERF, PRODUCTION.

Dans un premier temps, en tant que DevOps il nous faut livrer une application qui se déploie en fonction des environnements dans lesquels nous déployons l’application. Imaginons que nous ayons gardé l’inventory (fichier concerné par le tag « hosts » dans le playbook) du serveur de l’environnement de DEV, et que l’on veuille déployer l’application sur les serveurs de l’environnement d’intégration. Cela nous obligerait à changer les noms de serveurs à chaque déploiement. Imaginons que nous ayons laissé en dur le nom du serveur LDAP de développement dans notre code ANSIBLE, il faudrait changer cette valeur à chaque déploiement dans un autre environnement, puisque le serveur LDAP n’est sûrement pas le même.

ANSIBLE nous offre la possibilité de créer des applications, dont la configuration est entièrement paramétrable en fonction du contexte, c’est à dire l’environnement.

Notamment grâce aux variables, aux templates et aux rôles.

L’objectif de ce TP est de transformer les playbooks utilisés pour déployer l’application sample.war , en un seul playbook qui installera toute l’application en fonction des environnements.

Ce playbook devra appeler d’autres playbooks qui eux inclueront les rôles apache et tomcat, et deploy\_war.

Le répertoire contenant le playbook devra avoir la structure suivante :

<https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html#directory-layout>

inventories/

**prod**/

hosts

group\_vars/

group1.yml # here we assign variables to particular groups

group2.yml

host\_vars/

hostname1.yml # here we assign variables to particular systems

hostname2.yml

**dev**/

# hosts : inventory file for DEV servers  : Il devra contenir les groupes

# « webservers » and « appservers »

hosts

**group\_vars/**

# Les variables déclarées dans ce fichiers ne seront connues que des

# serveurs appartenant au group webservers

webservers.yml # here we assign variables to particular groups

# Les variables déclarées dans ce fichiers ne seront connues que des

# serveurs appartenant au group appservers

appservers.yml

**host\_vars/**

# Les variables déclarés dans le fichier webhost1.yml ne concerne que le

# server webhost1, et « webhost1 » doit figurer dans le fichier « dev/hosts »

webhost1.yml # here we assign variables to particular systems

# Les variables déclarés dans le fichier webhost1.yml ne concerne que le

# server webhost1, et « apphost1 » doit figurer dans le fichier « dev/hosts »

apphost1.yml

# Le fichier d’installation de l’application qui va appeler toutes les autres #

# playbooks

site.yml

# le playbook qui va appeler le rôle APACHE

webservers.yml

# le playbook qui va appeler le rôle TOMCAT

appservers.yml

# le playbook qui va appeler le rôle DEPLOY\_APP

deploy-sample-app.yml

# Le répertoire qui contient les rôles qui seront appelés par les playbooks

<https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html#top-level-playbooks-are-separated-by-role>

roles/

# Le rôle TOMCAT

tomcat/

# Le rôle TOMCAT

apache/

# Le rôle DEPLOY\_APP

deploy\_app/

Ansible exprime toute sa puissance dans cette flexibilité. Son mécanisme interne lui permet de d’appliquer la configuration selon le contexte de manière très fine.

En fonction de la valeur entrée dans le tag « hosts » (un groupe, un nom de serveur ou une IP), il sélectionnera automatiquement les bonnes valeurs des variables présentes dans les playbook, ou les rôles.

Exemple de fichier site.yml :

---

# file: site.yml

- import\_playbook: webservers.yml

- import\_playbook: appservers.yml

Exemple de fichier webservers.yml :

---

# file: webservers.yml

- hosts: webservers

roles:

- apache

Exemple de fichier appservers.yml :

---

# file: appservers.yml

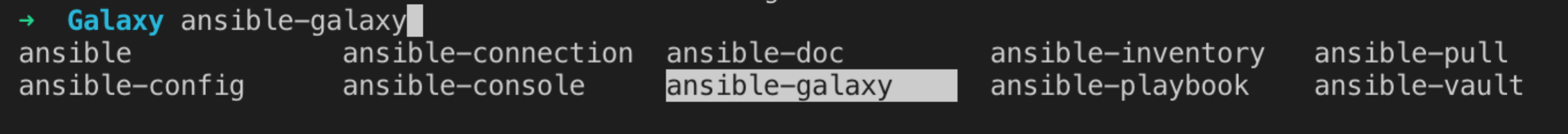
- hosts: appservers

roles:

- tomcat

- deploy\_app

Pour créer la structure de nos roles nous utilserons l’outil Ansible Galaxy qui est assez pratique : <https://galaxy.ansible.com/>

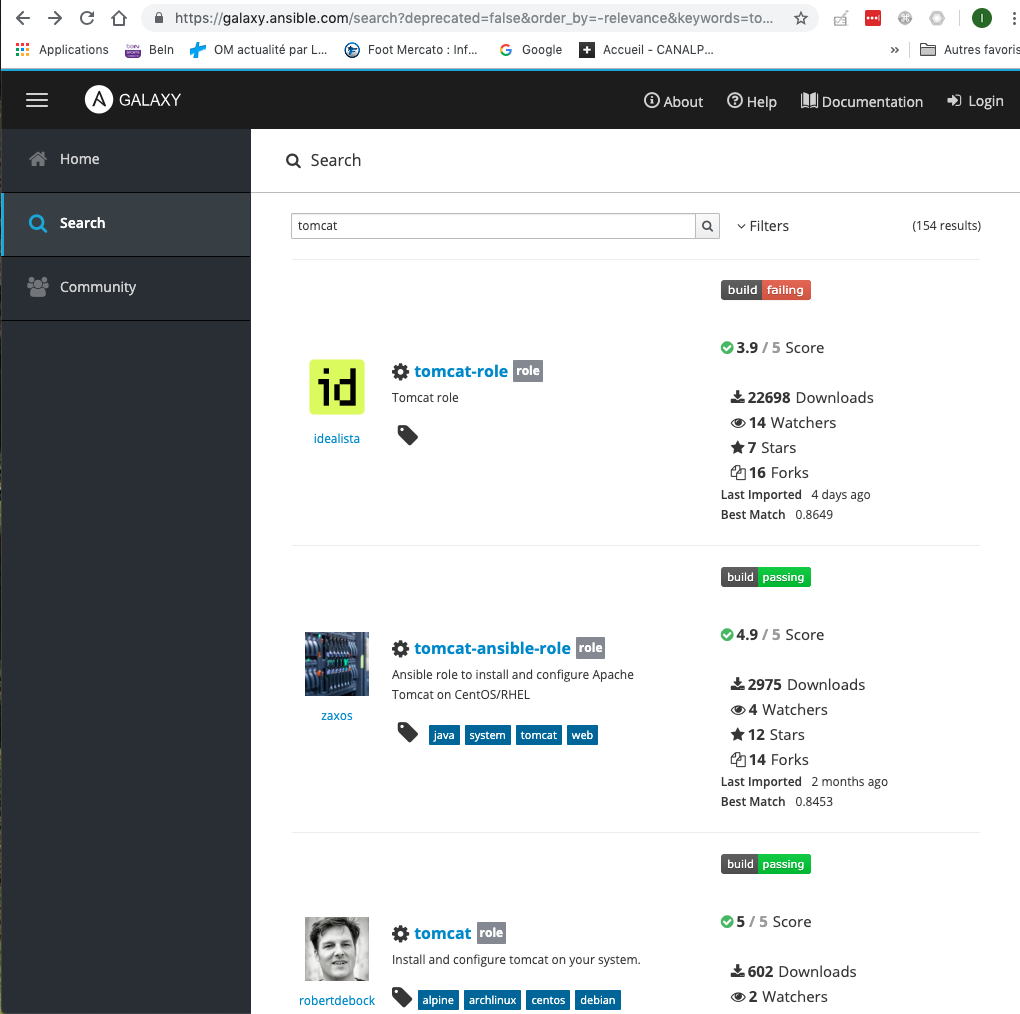


Vous trouverez la documentation complète ici: <https://docs.ansible.com/ansible/latest/reference_appendices/galaxy.html>

Sa principale fonctionnalité est qu’à l’instar de GitHub qui héberge du code, Ansible GALAXY héberge des rôles Ansible qui sont testés et notés par la communauté.

Rendez vous sur le site : <https://galaxy.ansible.com/>

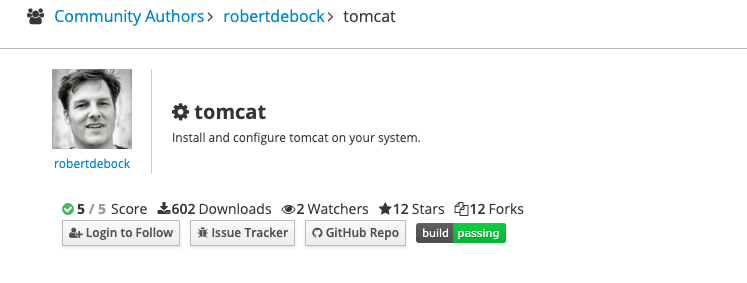
Cliquer sur le lien “Search” dans le panneau de gauche du navigateur.



Je clique sur GitHub Repo pour avoir accès à la documentation du rôle

<https://github.com/robertdebock/ansible-role-tomcat>

Et on peut dire qu’il ne blague pas quand il fait un role Ansible ☺



Il nous donne un exemple de playbook, parfait.

---

- name: Converge

hosts: all

become: yes

gather\_facts: yes

vars:

tomcat\_instances:

- name: "tomcat"

- name: "tomcat-version-7"

version: 7

shutdown\_port: 8007

non\_ssl\_connector\_port: 8082

ssl\_connector\_port: 8445

ajp\_port: 8011

- name: "tomcat-version-8"

version: 8

shutdown\_port: 8008

non\_ssl\_connector\_port: 8083

ssl\_connector\_port: 8446

ajp\_port: 8012

- name: "tomcat-version-9"

version: 9

shutdown\_port: 8019

non\_ssl\_connector\_port: 8084

ssl\_connector\_port: 8447

ajp\_port: 8013

- name: "tomcat-specific"

user: "specificuser"

group: "specificgroup"

shutdown\_port: 8020

non\_ssl\_connector\_port: 8085

ssl\_connector\_port: 8448

ajp\_port: 8014

xms: 256M

xmx: 512M

- name: "tomcat-with-wars"

shutdown\_port: 8021

non\_ssl\_connector\_port: 8086

ssl\_connector\_port: 8449

ajp\_port: 8015

wars:

- url: https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war

- name: "tomcat-java\_opts"

shutdown\_port: 8022

non\_ssl\_connector\_port: 8087

ssl\_connector\_port: 8449

ajp\_port: 8016

java\_opts:

- name: UMASK

value: "0007"

roles:

- robertdebock.tomca

Un peu de ménage :

---

- name: TOMCAT | INSTALL

hosts: all

become: yes

gather\_facts: yes

vars:

tomcat\_instances:

- name: "tomcat-with-wars"

version: 8

shutdown\_port: 8021

non\_ssl\_connector\_port: 8080

ssl\_connector\_port: 8443

ajp\_port: 8009

xms: 256M

xmx: 512M

wars:

- url: https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war

roles:

- robertdebock.tomcat

Je crèe un repertoire “roles”

➜ Galaxy ll

total 896

drwxr-xr-x 9 Loic staff 288 29 avr 20:31 ..

-rw-r--r--@ 1 Loic staff 88536 29 avr 21:54 Word Work File L\_365247426.tmp

-rw-r--r-- 1 Loic staff 449 29 avr 22:37 appservers.yml

-rw-r--r--@ 1 Loic staff 356571 29 avr 22:38 Ansible-galaxy-TP.docx

-rw-r--r-- 1 Loic staff 892 29 avr 22:50 Vagrantfile

drwxr-xr-x 5 Loic staff 160 29 avr 22:50 .vagrant

drwxr-xr-x 3 Loic staff 96 29 avr 22:57 **roles**

drwxr-xr-x 8 Loic staff 256 29 avr 22:58 .

➜ Galaxy

Je crèe le fichier “requirements.yml” dans le repertoire “roles”

# from galaxy

- src: robertdebock.tomcat

- src: robertdebock.service

Je choisi le troisième … ☺

<https://docs.ansible.com/ansible/latest/reference_appendices/galaxy.html#installing-multiple-roles-from-a-file>

Vagrantfile

Vagrant.configure("2") do |config|

# Linux OS CentOS

config.vm.box = "geerlingguy/centos7"

# Web server

config.vm.define "web-server" do |web|

web.vm.hostname = "devops-web.dev"

# static ip address

web.vm.network :private\_network, ip: "192.168.60.14"

web.vm.network :forwarded\_port, guest: 80, host: 8080

end

# Application server

config.vm.define "app-server" do |app|

app.vm.hostname = "devops-app.dev"

# static ip address

app.vm.network :private\_network, ip: "192.168.60.15"

app.vm.provision "playbook-app", type:'ansible' do |ansible|

ansible.playbook = "appservers.yml"

end

end

end

➜ Galaxy **ansible-galaxy install -r roles/requirements.yml**

[WARNING]: - robertdebock.tomcat (4.0.3) is already installed - use --force to change

version to unspecified

- downloading role 'service', owned by robertdebock

- downloading role from https://github.com/robertdebock/ansible-role-service/archive/2.0.3.tar.gz

- extracting robertdebock.service to /Users/Loic/.ansible/roles/robertdebock.service

- robertdebock.service (2.0.3) was installed successfully

appservers.yml

---

- name: TOMCAT | INSTALL

hosts: all

become: yes

gather\_facts: yes

vars:

tomcat\_instances:

- name: "tomcat-with-wars"

version: 8

shutdown\_port: 8021

non\_ssl\_connector\_port: 8080

ssl\_connector\_port: 8443

ajp\_port: 8009

xms: 256M

xmx: 512M

wars:

- url: https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war

roles:

- robertdebock.tomcat

➜ Galaxy **vagrant provision app-server**

==> app-server: Running provisioner: playbook-app (ansible)...

Vagrant has automatically selected the compatibility mode '2.0'

according to the Ansible version installed (2.7.5).

Alternatively, the compatibility mode can be specified in your Vagrantfile:

https://www.vagrantup.com/docs/provisioning/ansible\_common.html#compatibility\_mode

app-server: Running ansible-playbook...

PLAY [TOMCAT | INSTALL] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [app-server]

TASK [robertdebock.tomcat : loop over tomcat\_instances] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

included: /Users/Loic/.ansible/roles/robertdebock.tomcat/tasks/instance.yml for app-server

TASK [robertdebock.tomcat : ensure instance group exists] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [app-server]

TASK [robertdebock.tomcat : ensure instance user exists] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [app-server]

TASK [robertdebock.tomcat : ensure instance directory exists] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [app-server]

TASK [robertdebock.tomcat : install tomcat instance] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [app-server]

TASK [robertdebock.tomcat : save instance name] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [app-server]

TASK [robertdebock.tomcat : configure tomcat instance - server.xml] \*\*\*\*\*\*\*\*\*\*\*\*

changed: [app-server]

TASK [robertdebock.tomcat : configure tomcat instance - setenv.sh] \*\*\*\*\*\*\*\*\*\*\*\*\*

skipping: [app-server]

TASK [robertdebock.service : test veriables and environment] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [app-server] => {

"changed": false,

"msg": "All assertions passed"

}

TASK [robertdebock.service : test service\_list] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [app-server] => (item=tomcat-with-wars) => {

"changed": false,

"item": {

"description": "tomcat-with-wars",

"group\_name": "tomcat",

"name": "tomcat-with-wars",

"start\_command": "/opt/tomcat-with-wars/bin/catalina.sh run",

"user\_name": "tomcat"

},

"msg": "All assertions passed"

}

TASK [robertdebock.service : install required packages] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [app-server]

TASK [robertdebock.service : configure systemd service] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [app-server] => (item=tomcat-with-wars)

TASK [robertdebock.service : configure sysvinit service] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

skipping: [app-server] => (item=tomcat-with-wars)

RUNNING HANDLER [robertdebock.tomcat : systemctl daemon-reload] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [app-server]

RUNNING HANDLER [robertdebock.tomcat : restart tomcat instance] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [app-server]

TASK [robertdebock.tomcat : start and enable tomcat instance] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [app-server]

TASK [robertdebock.tomcat : loop over wars] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

included: /Users/Loic/.ansible/roles/robertdebock.tomcat/tasks/war.yml for app-server

TASK [robertdebock.tomcat : deploy war url] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [app-server]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

app-server : ok=17 changed=9 unreachable=0 failed=0

Cool ☺

On peut également créer le rôle Apache nous même.

➜ roles ansible-galaxy init apache --offline

➜ roles cd ..

➜ Galaxy cd roles

➜ roles ll

total 8

drwxr-xr-x 3 Loic staff 96 29 avr 23:03 .

drwxr-xr-x 9 Loic staff 288 29 avr 23:15 ..

-rw-r--r-- 1 Loic staff 68 29 avr 23:17 requirements.yml

➜ roles **ansible-galaxy init apache --offline**

- apache was created successfully

➜ roles tree apache

**apache**

├── README.md

├── defaults

│ └── main.yml

├── files

├── handlers

│ └── main.yml

├── meta

│ └── main.yml

├── tasks

│ └── main.yml

├── templates

├── tests

│ ├── inventory

│ └── test.yml

└── vars

└── main.yml

8 directories, 8 files

➜ roles

Les rôles sont essentiels dans la philosophie de l’outil.

La notion de « reusables » roles : <https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html>

* tasks - contains the main list of tasks to be executed by the role.
* handlers - contains handlers, which may be used by this role or even anywhere outside this role.
* defaults - default variables for the role (see [Using Variables](https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html) for more information).

<https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html#role-default-variables>

Les variables avec les plus basses priorités !!!

* vars - other variables for the role (see [Using Variables](https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html) for more information).
* files - contains files which can be deployed via this role.
* templates - contains templates which can be deployed via this role. JINJA2 : les fichiers de configuration d’Apache comme le worker.properties
* meta - defines some meta data for this role. See below for more detail

Je commence par copier le playbook install-webserver.yml que nous avons réalisé dans le TP précèdent, dans le répertoire: « roles/apache/tasks », et je le renomme bien sur en en install-apache.yml.

Puis je supprime les lignes suivantes :

- name: This sets up an httpd webserver

hosts: web-server

become: yes

tasks:

On obtient donc le fichier suivant :

roles/apache/tasks/install-apache.yml

---

- name: Install apache packages

yum:

name: "httpd,httpd-devel,apr,apr-devel,apr-util,apr-util-devel,gcc,gcc-c++,make,autoconf,libtool"

state: present

- name: ensure httpd is running

service:

name: httpd

state: started

Puis j’ajoute les lignes suivantes dans le fichier roles/apache/tasks/main.yml :

---

# tasks file for apache

- name: Install Apache

include: install-apache.yml

Et j’avais déjà donné la structure du playbook à lancer pour déployer le role apache :

---

# file: webservers.yml

- hosts: webservers

roles:

- apache

# ansible-playbook –i inventories/DEV/hosts webservers.yml