



DevOps Tools

Day - 12



Prakash Ramamurthy

prakash.ramamurthy@wipro.com

Agenda



Ansible Modules



Ansible Playbook



Ansible Roles



Hands-On Demonstration





Ansible Modules



Sensitivity: Internal & Restricted

DevOps Professional

Ansible

Ansible Modules

- Ansible provides a module library having extensive set of modules which can be directly executed on remote hosts or using Playbooks written in YAML format.
- Modules can also be customized or added by users. These modules can be used to install packages, start services, create files and folders.
- Each module is associated with arguments. Arguments allow for customization of use of modules for specific requirement.s



Ansible Playbook



Sensitivity: Internal & Restricted

DevOps Professional

Ansible

Ansible Playbooks

- Ansible activities are scripted in the form of playbook.
- These are used for configuration management, deployment and orchestration purpose
- Playbooks describe the way the desired state can be created on remote systems.
- Playbook provide the design plan, using modules as tools.
- Playbooks help sequencing multi-tier rollouts involving rolling updates
- Playbooks can interact with monitoring servers and load balancers and based on their information can delegate actions on other hosts.
- Playbooks are human-readable and are developed using YAML syntax.

DevOps Professional

Ansible

Ansible Playbooks

- Ansible playbooks are more powerful and provide a modular way to get desired state.
- Playbooks are written using YAML format. They use minimum of syntax and avoid the complexity of programming language or script.
- Playbooks provide a list of 'plays' to achieve the desired state
- Each play map a group of hosts with specific roles
- Roles are built with ansible modules called tasks.

DevOps Professional

Ansible

Ansible Tasks

- Tasks are executed in the order they are provided in the playbook.
- Once pushed tasks are executed concurrently on all machines. Matching of tasks to machines is done by playbook.
- If any of the tasks are failed on a particular host, then rest of the tasks on the same host will not be executed.
- Each task execute a module associated with specific arguments.
- Each module work to bring the system to the desired state.
- If a playbook runs multiple times, it may not make any change unless required to bring the machine to desired state.

DevOps Professional

Ansible

Ansible Tasks

- Each task is associated with a name.
- Name is displayed while the task is running
- This output is for human observers to check whether the task executed successfully.
- Tasks use “module: options” format.
- Example:

```
---
- hosts: all
  tasks:
    - name: Create file
      file:
        path: /root/foo.conf
        state: touch
        mode: "u=rw,g=r,o=r"
```

DevOps Professional

Ansible

Ansible Handlers

- Handlers are similar to tasks
- Handlers are referenced by some other task by their name.
- Tasks notify the handlers and invoke their execution
- If the handlers are not notified by the tasks, then they will not run.
- Tasks trigger these 'notify' actions at the end of each block of tasks in playbook
- Example

```
handlers:
```

```
- name: Start Apache Service
  service: name=httpd state=started
```

DevOps Professional

Ansible

Ansible Variables

- Variable definition allows uniform use of numbers, paths and other quantities across playbook

```
- hosts: webservers
  vars:
    http_port: 80
```

- Check the documentation to know rules to define variables.
- Ansible variables can be referenced in tasks, handlers, templates.
- Refer Ansible variable as below:

```
My amp goes to {{ max_amp_value }}
```

DevOps Professional

Ansible

Ansible Templates

- Templates are the way to take care of variations across servers
- Templates are in Jinja2 template language (<http://jinja.pocoo.org/docs/>)
- Below is an example of template in J2 format:

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerName {{ item.servername }}
    DocumentRoot /var/www/html/{{ item.documentrootdir }}
    ServerAdmin {{ item.serveradmin }}
    ErrorLog logs/{{ item.errorlog }}-error_log
    CustomLog logs/{{ item.customlog }}-access_log common

    <Directory /var/www/html/{{ item.documentrootdir }}>
        Options None
    </Directory>

{% include "files/" + item.documentrootdir + ".conf" ignore missing %}

</VirtualHost>
```

DevOps Professional

Ansible

Ansible for Automation

- Heterogeneity among servers is the biggest challenge for automation.
- Preserving the servers as they are automation should adopt the script to different servers
- Templates help adoption to different servers, Variables maintain uniformity of values across playbook.
- Often automation need to adopt to environment at run time
- Gathering Facts from target server help variables and templates to make changes to playbook operations.



Ansible Roles



Sensitivity: Internal & Restricted

DevOps Professional

Ansible

Ansible Roles

- Roles provide a structured organization to playbooks
- Roles allow tasks, handlers, templates, variables to be placed in separate files in separate folders
- Further they can also be placed into multiple files for easy management.
- Playbooks can also use plays from other playbooks and dependencies can be created.
- Roles would allow minor routine plays to be written in smaller files and by including them in other yaml files allow abstraction of activities for better manageability.

DevOps Professional

Ansible

Ansible Roles

- Example of roles file structure is on the right side of slide
- Invoking of roles in playbooks is given below

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

```
site.yml  
webservers.yml  
fooservers.yml  
roles/  
  common/  
    files/  
    templates/  
    tasks/  
    handlers/  
    vars/  
    meta/  
  webservers/  
    files/  
    templates/  
    tasks/  
    handlers/  
    vars/  
    meta/
```


DevOps Professional

Ansible

Ansible Roles

This designates the following behaviors, for each role 'x':

- Tasks in the roles : roles/x/tasks/main.yml
- Handlers in the roles : roles/x/handlers/main.yml
- Variables in the roles : roles/x/vars/main.yml
- Role dependencies : roles/x/meta/main.yml
- Files required by copy tasks : roles/x/files/
- Template files : roles/x/templates/

DevOps Professional

Ansible

Ansible Galaxy

- Ansible Galaxy is the community site providing Ansible Roles for various activity.
<https://galaxy.ansible.com/>

To download Ansible roles from above site you may use the below command.

```
ansible-galaxy install username.rolename
```

- Create basic file structure for roles.

```
ansible-galaxy init rolename
```



Hands-On Demonstration



Sensitivity: Internal & Restricted

Handson Demonstration

Pushing a module

- Pushing a command module

```
$ ansible localhost -m command -a 'echo "Hello World"'
```

```
$ ansible localhost -a 'echo "Hello World"'
```

- Pushing a shell module

```
$ ansible localhost -m shell -a 'cat /etc/resolv.conf'
```

- Using touch shell module to create file

```
$ ansible localhost -m shell -a 'touch /home/osgdev/ansilab/testfile'
```

Handson Demonstration

Creating a file

- Using file module

```
$ ansible localhost -m file -a 'path=/home/osgdev/ansilab/testfile state=touch'
```

- Check for file availability

```
$ ansible localhost -m file -a 'path=/home/osgdev/ansilab/testfile state=file'
```

- Creating Directory

```
$ ansible localhost -m file -a 'path=/home/osgdev/ansilab/resource  
state=directory'
```

- Removing File

```
$ ansible localhost -m file -a 'path=/home/osgdev/ansilab/testfile state=absent'
```

- Removing Directory

```
$ ansible localhost -m file -a 'path=/home/osgdev/ansilab/resource state=absent'
```

Handson Demonstration

Working on file

- Copying the file

```
$ ansible localhost -m copy -a 'src=/home/osgdev/ansilab/ansible.log  
dest=/home/osgdev/ansilab/ansible1.txt'
```

- Unarchiving the file

```
$ ansible localhost -m unarchive -a 'src=/home/osgdev/myDownloads/apache-tomcat-  
8.5.27.tar.gz'
```

Handson Demonstration

Working on Tomcat

- Unarchiving tomcat tar file

```
$ ansible localhost -m unarchive -a 'src=/home/osgdev/myDownloads/apache-tomcat-8.5.27.tar.gz'
```

- Changing server.xml file for tomcat connector port

```
$ ansible localhost -m copy -a 'src=/home/osgdev/resource/apache-tomcat-8.5.27/conf/server.xml dest=/home/osgdev/resource/server.xml.j2'
```

```
$ ansible localhost -m copy -a 'src=/home/osgdev/resource/server.xml.j2 dest=/home/osgdev/resource/apache-tomcat-8.5.27/conf/server.xml'
```

- Starting tomcat server

```
$ ansible localhost -m shell -a '/home/osgdev/resource/apache-tomcat-8.5.27/bin/startup.sh'
```

Handson Demonstration

Creating Playbook for command and shell modules

echo.yaml

```
---
- hosts: localhost
  tasks:
    - name: Use echo command
      command: 'echo "Hello world"'
```

\$ **ansible-playbook -v echo.yaml**

hostname.yaml

```
- hosts: localhost
  tasks:
    - name: Use hostname command
      command: hostname
```

\$ **ansible-playbook -v hostname.yaml**

resolv.yaml

```
- hosts: localhost
  tasks:
    - name: Use shell command
      shell: cat /etc/resolv.conf
```

\$ **ansible-playbook resolv.yaml**

Handson Demonstration

Creating a folder with a file inside

```
- hosts: localhost
  tasks:
    - name: To Create a folder
      file:
        path: /home/osgdev/ansilab/NEWSAMPLE
        state: directory
        mode: 0755

    - name: To Create a file
      file:
        path: /home/osgdev/ansilab/NEWSAMPLE/new_file
        state: touch

$ ansible-playbook file.yaml
```

Handson Demonstration

Use of variable in playbook

```
- hosts: localhost                                $ ansible-playbook file1.yaml

vars:
  folder_path: /home/osgdev/ansilab

tasks:
- name: To Create a folder
  file:
    path: "{{folder_path}}/NEWSAMPLE2"
    state: directory
    mode: 0755

- name: To Create a file
  file:
    path: "{{folder_path}}/NEWSAMPLE2/new_file2"
    state: touch
```



Handson Demonstration

Use of handlers in playbook

```
- hosts: localhost
```

```
$ ansible-playbook file2.yaml
```

```
vars:
```

```
  folder_path: /home/osgdev/ansilab
```

```
tasks:
```

```
- name: To Create a folder
```

```
  file:
```

```
    path: "{{folder_path}}/NEWSAMPLE3"
```

```
    state: directory
```

```
    mode: 0755
```

```
  notify:
```

```
    - To Create a file
```

```
handlers:
```

```
- name: To Create a file
```

```
  file:
```

```
    path: "{{folder_path}}/NEWSAMPLE3/new_file3"
```

```
    state: touch
```

Handson Demonstration

Creation of Role

- Roles path in ansible.cfg file:
[defaults]

```
inventory = /home/osgdev/ansilab/ansiserver
log_path = /home/osgdev/ansilab/ansible.log
roles_path = /home/osgdev/ansilab/roles
```

- Folder Structure for Roles:

```
roles
├── sample
│   ├── handlers
│   ├── tasks
│   └── vars
```

Handson Demonstration

Vars, Tasks and Handlers in the Role

\$ `ansible-playbook server.yaml`

tasks/main.yaml

```
- name: To Create a folder
  file:
    path: "{{folder_path}}/NEWSAMPLE4"
    state: directory
    mode: 0755
  notify:
    - To Create a file
```

vars/main.yaml

```
folder_path: /home/osgdev/ansilab
```

server.yaml

```
- hosts: localhost
  roles:
    - sample
```

handlers/main.yaml

```
- name: To Create a file
  file:
    path: "{{folder_path}}/NEWSAMPLE4/new_file4"
    state: touch
```

Handson Demonstration

Role creation using galaxy

```
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
├── vars
│   └── main.yml
```

```
$ ansible-galaxy init ./roles/newrole
```



Thank You