Getting started with Ansible-playbook

- Ravdeep S Pasricha
DevOps Engineer
@ravdeep003

Agenda

- What is Ansible
- Demo
- Inventory
- Playbook
- Modules
- Variables
- Directory Structure

Ansible

- Ansible is an IT automation tool.

- It can:
  - configure systems,
  - deploy software, and
  - orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates.

- Ansible manages machines in an agentless manner.
- Uses OpenSSH for transport
- Highly scalable
- Uses YAML – Friendly Language
- Idempotent

DEMO & EXAMPLE

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum: pkg=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 (and enable it at boot)
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```

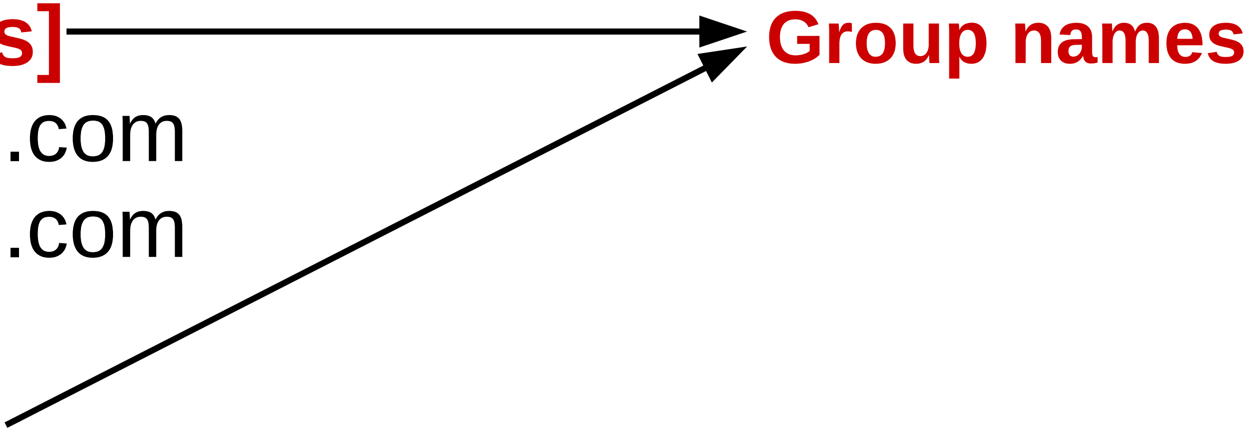
Inventory

- **Hosts and groups**:

```
mail.example.com

[webservers]
foo.example.com
bar.example.com

[dbservers]
one.example.com
two.example.com
three.example.com
```



Group names

Host_vars:

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

Group_vars:

```
[atlanta]
host1
host2

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

Playbooks

- Playbooks are Ansible’s configuration, deployment, and orchestration language
- Playbooks are expressed in YAML format
- The goal of a play is to map a group of hosts to some well defined roles, represented by things ansible calls tasks

Modules

- Apt
- Copy
- Template
- Service
- Shell

Variables

- extra vars (-e in the command line) always win
- kconnection variables defined in inventory (ansible_ssh_user, etc)
- then comes "most everything else" (command line switches, vars in play, included vars, role vars, etc)
- then comes the rest of the variables defined in inventory
- then comes facts discovered about a system
- then "role defaults", which are the most "defaulty" and lose in priority to everything.

Directory Structure

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

In a playbook, it would look like this:

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