

```
from visualapp import app as application
```

Refer the template `visualapp.wsgi.j2` in `visualapp_app` role's `tasks/main.yml`

```
# vi roles/visualapp_app/tasks/main.yml
```

```
copy: src=visualapp/app/ dest=/var/www/visualapp mode=0755
  notify: restart apache2

+- name: copy visualapp.wsgi
+ template: src=visualapp.wsgi.j2 dest=/var/www/visualapp/visualapp.wsgi mode=0755
+ notify: restart apache2
+
- name: copy apache virtual host config
  copy: src=visualapp/visualapp.conf dest=/etc/apache2/sites-available mode=0755
  notify: restart apache2
```

Put the variables name in `webserver.yml` playbook where we refer the `nginx` role

```
# vi webserver.yml
become: true
roles:
  - apache2
  - visualapp_app
+  - { role: visualapp_app, db_user: visualapp, db_pass: visualapp, db_name: visualapp }

# ansible-playbook webserver.yml
```

#### **Variables: `vars_files`, `group_vars` | Use `group_vars/all` file to specify variables for database**

Database variables are referred in two places

`visualapp.wsgi.j2` template from `visualapp_app` role &  
`tasks/main.yml` from `mysql` role.

We will use `group_vars/all` file to specify below variables

```
db_name: visualapp
db_user: visualapp
db_pass: visualapp
```

```
# mkdir group_vars
# vi group_vars/all
```

```
+++
+db_name: visualapp
+db_user: visualapp
+db_pass: visualapp
```

Remove db variables from `websever.yml` playbook

#### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```
# vi webserver.yml
```

```
become: true
roles:
  - apache2
- - { role: visualapp_app, db_user: visualapp, db_pass: visualapp, db_name: visualapp }
+ - visualapp_app
```

Refer db variables from group\_vars/all file inside database.yml playbook also called as variable routing.

Variable routing can be defined as one variable pointing to another variable.

E:g db\_user\_name: "{{ db\_user }}"

```
# vi database.yml
```

```
- hosts: database
become: true
roles:
- - { role: mysql, db_name: visualapp, db_user_name: visualapp, db_user_pass: visualapp,
db_user_host: '%' }
+ - role: mysql
+   db_user_name: "{{ db_user }}"
+   db_user_pass: "{{ db_pass }}"
+   db_user_host: '%'
```

### Variables: vault | encrypting db\_pass variables value with vault

[https://docs.ansible.com/ansible/playbooks\\_vault.html](https://docs.ansible.com/ansible/playbooks_vault.html)

Instead of having all file for variables, we will create all directory and move our variables in all directory.

```
# cd group_vars
# mv all vars
# mkdir all
# mv vars all
# export EDITOR=vim
```

Change directory to all directory and create vault. Vault file should be present in all directory for our scenario.

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```
# cd all  
# ansible-vault create vault  
Give a vault password and put below mentioned content.
```

```
+ ---  
+ vault_db_pass : admin123
```

Refer to vault\_db\_pass in all/vars file

```
# vi vars  
+---  
+db_name: visualapp  
+db_user: visualapp  
+db_pass: "{{ vault_db_pass }}"
```

Execute database.yml playbook, you should get some error like below

```
# ansible-playbook database.yml
```

```
ERROR! Decryption failed  
ERROR! A vault password must be specified
```

Either you can give –ask-vault-pass option and execute the playbook which will ask you the vaultpassword or you can use a file where you specify vault password.

“vaultpass” is our vault password.

```
# echo "vaultpass" > ~/.vault_pass.txt  
# chmod 0600 ~/.vault_pass.txt  
# vi ansible.cfg  
[defaults]  
inventory = ./dev  
+vault_password_file = ~/.vault_pass.txt
```

Execute database playbook to test it

```
# ansible-playbook database.yml
```

Login to db01 instance and login to mysql database to verify.

```
# ssh db01  
# mysql -h localhost -u visualapp -p
```

## Advanced Execution

**Advanced Execution: gather\_facts | By disabling gather\_facts we can save the execution time.**

Time the execution with time command for site.yml, stack\_status and stack\_restart.yml

```
# time ansible-plabook site.yml  
# time ansible-plabook stack_status.yml  
# time ansible-plabook stack_restart.yml
```

Disable gather\_facts for all the plays except dbserver because we use fact varible for dbservers play.

```
ansible_eth1.ipv4.address
```

```
control.yml
```

```
---  
- hosts: control  
  become: true  
+ gather_facts: false  
  roles:  
    - control
```

```
loadbalancer.yml
```

```
---  
- hosts: loadbalancer  
  become: true  
+ gather_facts: false  
  roles:  
    - nginx
```

```
hostname.yml
```

```
---  
- hosts: all  
+ gather_facts: false  
  tasks:  
    - name: get server hostname  
      command: hostname
```

```
stack_restart.yml
```

```
# Bring stack down  
- hosts: loadbalancer  
  become: true  
+ gather_facts: false
```

**Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```

tasks:
  - service: name=nginx state=stopped
  - wait_for: port=80 state=drained

- hosts: webserver
  become: true
+ gather_facts: false
* tasks:
  - service: name=apache2 state=stopped
  - wait_for: port=80 state=stopped

# Bring stack up
- hosts: webserver
  become: true
+ gather_facts: false
tasks:
  - service: name=apache2 state=started
  - wait_for: port=80

- hosts: loadbalancer
  become: true
+ gather_facts: false
tasks:
  - service: name=nginx state=started
  - wait_for: port=80

```

stack\_status.yml

```

---
- hosts: loadbalancer
  become: true
+ gather_facts: false
tasks:
  - name: verify nginx service
    command: service nginx status

- hosts: webserver
  become: true
+ gather_facts: false
tasks:
  - name: verify apache2 service
    command: service apache2 status

  wait_for: host={{ ansible_eth0.ipv4.address }} port=3306 timeout=1

- hosts: control
+ gather_facts: false
tasks:
  - name: verify end-to-end index response

```

#### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```

uri: url=http://{{item}} return_content=yes
with_items: "{{lb_db.results}}"

- hosts: loadbalancer
+ gather_facts: false
  tasks:
    - name: verify backend index response
      uri: url=http://{{item}} return_content=yes

```

webserver.yml

```

---
- hosts: webserver
  become: true
+ gather_facts: false
  roles:
    - apache2

```

Time the execution again with time command to verify.

### **Extracting Repetitive Tasks: cache\_valid\_time**

Open site.yml and add below mentioned play

```

---
+- hosts: all
+ become: true
+ gather_facts: false
+ tasks:
  - name: update apt cache
+   apt: update_cache=yes cache_valid_time=86400
+
- include: control.yml
- include: database.yml
- include: webserver.yml

```

Remove update\_cache=yes parameter from all the tasks/main.yml of all the roles.

# vi roles/mysql/tasks/main.yml

E:g

```

- apt: name=mysql-server state=present update_cache=yes
+ apt: name=mysql-server state=present

```

## **Limits Execution by Hosts : limit**

```
# ansible-playbook site.yml --limit app01
```

## **Limits Execution by Tasks : tags**

We can also select particular task or tasks by tagging them and then using tag name while executing playbooks or site.yml.

```
# vi roles/control/tasks/main.yml
with_items:
  - apache2
  - libapache2-mod-wsgi
+ tags: [ 'packages' ]
```

List the available tags.

```
# $ ansible-playbook site.yml --list-tags
```

Execute it to verify

```
# ansible-playbook site.yml --tags "packages"
```

We can also skip the specific tags and execute the rest.

```
# $ ansible-playbook site.yml --skip-tags "packages"
```

Tag all the tasks as per our 4 principles of app deployment on Linux systems

```
['packages']
['service']
['system']
['configure']
```

```
# vi roles/apache2/tasks/main.yml
```

```
with_items:
  - apache2
  - libapache2-mod-wsgi
+ tags: [ 'packages' ]

  - name: ensure mod_wsgi enabled
    apache2_module: state=present name=wsgi
    notify: restart apache2
+ tags: [ 'system' ]

  - name: de-activate default apache site
    file: path=/etc/apache2/sites-enabled/000-default.conf state=absent
    notify: restart apache2
+ tags: [ 'system' ]

  - name: ensure apache2 started
```

```

service: name=apache2 state=started enabled=yes
+ tags: [ 'service' ]

# vi roles/control/tasks/main.yml

with_items:
  - curl
  - python-httplib2
+ tags: [ 'packages' ]

# vi roles/visualapp_app/tasks/main.yml

- python-pip
- python-virtualenv
- python-mysqldb
+ tags: [ 'packages' ]

- name: copy visualapp app source
  copy: src=visualapp/app/ dest=/var/www/visualapp mode=0755
  notify: restart apache2
+ tags: [ 'configure' ]

- name: copy visualapp.wsgi
  template: src=visualapp.wsgi.j2 dest=/var/www/visualapp/visualapp.wsgi mode=0755
  notify: restart apache2
+ tags: [ 'configure' ]

- name: copy apache virtual host config
  copy: src=visualapp/visualapp.conf dest=/etc/apache2/sites-available mode=0755
  notify: restart apache2
+ tags: [ 'configure' ]

- name: setup python virtualenv
  pip: requirements=/var/www/visualapp/requirements.txt virtualenv=/var/www/visualapp/.venv
  notify: restart apache2
+ tags: [ 'system' ]

- name: activate visualapp apache site
  file: src=/etc/apache2/sites-available/visualapp.conf dest=/etc/apache2/sites-enabled/visualapp.conf state=link
  notify: restart apache2
+ tags: [ 'configure' ]

# vi roles/mysql/tasks/main.yml

apt: name={{item}} state=present
with_items:
  - python-mysqldb
+ tags: [ 'packages' ]

```

#### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```

- name: install mysql-server
  apt: name=mysql-server state=present
+ tags: [ 'packages' ]

- name: ensure mysql listening on all ports
  lineinfile: dest=/etc/mysql/my.cnf regexp=^bind-address
    line="bind-address = {{ ansible_eth0.ipv4.address }}"
  notify: restart mysql
+ tags: [ 'configure' ]

- name: ensure mysql started
  service: name=mysql state=started enabled=yes
+ tags: [ 'service' ]

- name: create database
  mysql_db: name={{ db_name }} state=present
+ tags: [ 'configure' ]

- name: create user
  mysql_user: name={{ db_user_name }} password={{ db_user_pass }} priv={{ db_name }}.*:ALL
    host='{{ db_user_host }}' state=present
+ tags: [ 'configure' ]

# vi roles/nginx/tasks/main.yml

apt: name={{item}} state=present
with_items:
  - python-httplib2
+ tags: [ 'packages' ]

- name: install nginx
  apt: name=nginx state=present
+ tags: [ 'packages' ]

- name: configure nginx sites
  template: src=nginx.conf.j2 dest=/etc/nginx/sites-available/{{ item.key }} mode=0644
  with_dict: sites
  notify: restart nginx
+ tags: [ 'configure' ]

- name: get active sites
  shell: ls -1 /etc/nginx/sites-enabled
  register: active
+ tags: [ 'configure' ]

- name: de-activate sites
  file: path=/etc/nginx/sites-enabled/{{ item }} state=absent
  with_items: active.stdout_lines
  when: item not in sites
  notify: restart nginx

```

#### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```

+ tags: [ 'configure' ]

- name: activate nginx sites
  file: src=/etc/nginx/sites-available/{{ item.key }} dest=/etc/nginx/sites-enabled/{{ item.key }}
state=link
  with_dict: sites
  notify: restart nginx
+ tags: [ 'configure' ]

- name: ensure nginx started
  service: name=nginx state=started enabled=yes
+ tags: [ 'service' ]

# vi site.yml

tasks:
  - name: update apt cache
    apt: update_cache=yes cache_valid_time=86400
+ tags: [ 'packages' ]

  - include: control.yml
  - include: database.yml.

```

Save execution time by skipping packages tags

```
# time $ ansible-playbook site.yml --skip-tags "packages"
```

### Troubleshooting, Testing & Validation ordering problem

Make sure to have mysql service start task after you changed the mysql configuration.

If there is some typo or bad config in my.cnf, notify mysql restart is going to stop mysql and while starting it will fail. If you run the playbook again first it will hit mysql start task which will fail to start the mysql service and your play is not going to reach to the task where it pushes new my.cnf file to fix the issue. Ordering is very important in such scenario.

```

- name: Install mysql-server
  apt: name=mysql-server state=present

```

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```
- name: ensure mysql listening on all ports
  lineinfile: dest=/etc/mysql/my.cnf regexp="^bind-address"
    line="bind-address = {{ansible_eth1.ipv4.address}}"

  notify: restart mysql

- name: Ensure Mysql started
  service: name=mysql state=started enabled=yes
```

### **Jumping to Specific tasks: list-tasks, step, start-at-task**

[https://docs.ansible.com/ansible/playbooks\\_startnstep.html](https://docs.ansible.com/ansible/playbooks_startnstep.html)

```
# ansible-playbook site.yml --step
```

Select y for yes, n for no and c to continue with the execution without asking the prompt.

List all the tasks from the playbook and select the task from where you want to start the execution.

```
# ansible-playbook site.yml --list-tasks

# ansible-playbook site.yml --start-at-task "copy visualapp app source"
```

### **Retrying failed hosts**

```
# ansible-playbook site.yml --limit @/home/ansible/site.retry
```

### **Syntax-Check & Dry-Run: syntax-check, check**

```
# ansible-playbook --syntax-check site.yml
```

```
# ansible-playbook --check site.yml
```

## **31. WordPress setup with Ansible**

We are going to learn how to automate the deployment of a LAMP stack and install WordPress. LAMP stands for Linux, Apache (a Web server), MySQL (a database) and PHP (server-side scripting). It is a technology stack on which you can deploy different Web applications. We are also going to explore the installation of WordPress, which is free and open source software for creating websites and blogs.

### **Assumptions:**

#### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

- Ansible installed on the server and target node is a ubuntu server.
- Inventory created with target server hostname/IP and credentials setup.
- ansible.cfg with inventory path host\_key\_checking taken care.

## Apache setup Playbook

Installs apache after apt update, starts apache service and wait for port 80 to come up.

```
---
- name: Install Apache web server
hosts: ubuntu
become: yes
become_method: sudo
gather_facts: true
tags: [web]
tasks:
- name: Update the software package repository
apt:
update_cache: yes
- name: Install Apache
package:
name: "{{ item }}"
state: latest
with_items:
- apache2
- wait_for:
port: 80
```

## Execute playbook

\$ ansible-playbook apache.yml -K

SUDO password:

```
PLAY [Install Apache web server] ****
TASK [setup] ****
ok: [ubuntu]
TASK [Update the software package repository] ****
changed: [ubuntu]
TASK [Install Apache] ****
changed: [ubuntu] => (item='apache2')
TASK [wait_for] ****
ok: [ubuntu]
PLAY RECAP ****
ubuntu : ok=4 changed=2 unreachable=0 failed=0
```

## Installing MySQL

Install the MySQL database server.

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```
---
- name: Install MySQL database server
hosts: ubuntu
become: yes
become_method: sudo
gather_facts: true
tags: [database]
tasks:
- name: Update the software package repository
apt:
update_cache: yes
- name: Install MySQL
package:
name: "{{ item }}"
state: latest
with_items:
- mysql-server
- mysql-client
- python-mysqldb
- name: Start the server
service:
name: mysql
state: started
- wait_for:
port: 3306
- mysql_user:
name: guest
password: '*F7B659FE10CA9FAC576D358A16CC1BC646762FB2'
encrypted: yes
priv: '.*:ALL,GRANT'
state: present
```

### **Execute Mysql Playbook.**

```
$ ansible-playbook mysql.yml -K
SUDO password:
PLAY [Install MySQL database server] ****
TASK [setup] ****
ok: [ubuntu]
TASK [Update the software package repository] ****
changed: [ubuntu]
TASK [Install MySQL] ****
changed: [ubuntu] => (item=[u'mysql-server', u'mysql-client', u'python-mysqldb'])
TASK [Start the server] ****
ok: [ubuntu]
TASK [wait_for] ****
ok: [ubuntu]
TASK [mysql_user] ****
ok: [ubuntu]
PLAY RECAP ****
ubuntu : ok=6 changed=2 unreachable=0 failed=0
```

## Installing PHP

PHP is a server-side programming language

### Installing PHP

```
---
- name: Install PHP
hosts: ubuntu
become: yes
become_method: sudo
gather_facts: true
tags: [web]
tasks:
- name: Update the software package repository
apt:
update_cache: yes
- name: Install PHP
package:
name: "{{ item }}"
state: latest
with_items:
- php5
- php5-mysql
```

```
$ ansible-playbook php.yml -K
```

```
SUDO password:
```

```
PLAY [Install PHP] ****
```

```
TASK [setup] ****
```

```
ok: [ubuntu]
```

```
TASK [Update the software package repository] ****
```

```
changed: [ubuntu]
```

```
TASK [Install PHP] ****
```

```
changed: [ubuntu] => (item=[u'php5', u'php5-mysql'])
```

```
PLAY RECAP ****
```

```
ubuntu : ok=3 changed=2 unreachable=0 failed=0
```

## Installing WordPress

Install and set up WordPress.

```
---
- name: Setup WordPress
hosts: ubuntu
become: yes
become_method: sudo
gather_facts: true
```

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```

tags: [database]
vars:
wordpress_file: "/home/{{ ansible_user }}/Downloads/wordpress-latest.zip"
wordpress_dest: "/var/www/html"
tasks:
- name: Update the software package repository
apt:
update_cache: yes
- name: Create a database for wordpress
mysql_db:
name: wordpress
state: present
- name: Create downloads directory
file:
path: "/home/{{ ansible_user }}/Downloads"
state: directory
- name: Create target directory
file:
path: "{{ wordpress_dest }}/wordpress"
state: directory
- name: Download latest wordpress
get_url:
url: https://wordpress.org/latest.zip
dest: "{{ wordpress_file }}"
- name: Extract to /var/www/html
unarchive:
src: "{{ wordpress_file }}"
dest: "{{ wordpress_dest }}"
remote_src: True
- name: Copy wp-config-sample.php to wp-config.php
command: cp "{{ wordpress_dest }}/wordpress/wp-config-sample.php" "{{ wordpress_dest }}/wordpress/wp-config.php"
- name: Update database credentials in the file
replace:
dest: "{{ wordpress_dest }}/wordpress/wp-config.php"
regexp: "{{ item.regexp }}"
replace: "{{ item.replace }}"
with_items:
- { regexp: 'database_name_here', replace: 'wordpress' }
- { regexp: 'username_here', replace: 'guest' }
- { regexp: 'password_here', replace: 'osfy' }
- name: Restart apache2 server
service:
name: apache2
state: restarted

```

```

$ ansible-playbook wordpress.yml -K
SUDO password:
PLAY [Setup WordPress] ****
TASK [setup] ****
ok: [ubuntu]
TASK [Update the software package repository] ****
changed: [ubuntu]
TASK [Create a database for wordpress] ****
changed: [ubuntu]
TASK [Create downloads directory] ****

```

#### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```

ok: [ubuntu]
TASK [Create target directory] *****
changed: [ubuntu]
TASK [Download latest wordpress] *****
ok: [ubuntu]
TASK [Extract to /var/www/html] *****
changed: [ubuntu]
TASK [Copy wp-config-sample.php to wp-config.php] *****
changed: [ubuntu]
TASK [Update database credentials in the file] *****
changed: [ubuntu] => (item={u'regexp': u'database_name_here', u'replace': u'wordpress'})
changed: [ubuntu] => (item={u'regexp': u'username_here', u'replace': u'guest'})
changed: [ubuntu] => (item={u'regexp': u'password_here', u'replace': u'osfy'})
TASK [Restart apache2 server] *****
changed: [ubuntu]
PLAY RECAP *****
ubuntu : ok=10 changed=7 unreachable=0 failed=0

```

If you open the URL `http://targetserverip/wordpress` in a browser on the host system, you will see wordpress setup page.

### **Summary:**

- ✓ Configuration Management tools have majorly replaced the scripting languages in DevOps for automation.
- ✓ It's easy to manage and automate infrastructure from a centralised place compared to scripts.
- ✓ Ansible is the current hot favourite automation tool and is very widely adopted in DevOps life cycles.
- ✓ It's easy to setup. Easy to read and write, does not need any programming knowledge.
- ✓ Ansible gives it little but very powerful AdHoc commands which is used to execute single tasks on multiple servers.
- ✓ Playbooks are automation scripts of Ansible and includes series of tasks that should be executed on selected hosts.
- ✓ Ansible Roles gives us a very modular structure to our code and make the code sharable with other team or projects. It focuses on reusability of the code.
- ✓ There are so many Sample Playbook in this chapter which you can use in your day to day DevOps Automation.
- ✓ Ansible is very innovative and releases new modules every now and then. Always refer Ansible documentation, because that's the single source of truth for everything in Ansible.

<https://docs.ansible.com/>

### **Conclusion:**

#### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

Practice as much as possible when it comes to Ansible, always read Ansible documentation. Its currently most widely used automation tool and you should be an expert of it if you are using it in DevOps lifecycle or any automation.



# XIII. Puppet

## 1. What is Puppet?

Puppet is a configuration management tool to automate infrastructure management and configuration i.e., it manages configuration data on other systems, including users, packages, processes, services. It helps in the concept of Infrastructure as code. Puppet written in Ruby DSL language, which can be easily managed and configured.

Puppet follows client-server Model, where one machine acts as server known as puppet master and the other acts as client known as slave or agent machine.

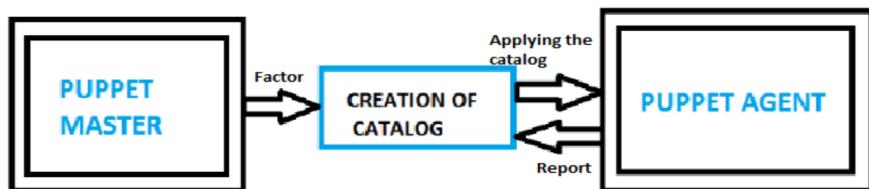
The Puppet Master is a Machine where all manifests will be developed and ready to be implemented on the agents.

The agent implements Puppet manifests, or files containing Puppet configuration language that declare the desired state of the node.

### Special Features and Work Flow

in Puppet, one can safely run the same set of configuration multiple times on the same machine. In this flow, Puppet checks for the status of the target machine and will only make changes when there is any specific change in the configuration.

#### Puppet Work Flow



Puppet architecture mainly contains following components.

#### **Puppet Master**

Puppet Master is the key mechanism which handles all the configuration related stuff. It applies the configuration to nodes using the Puppet agent.

#### **Catalog**

All the configuration changes which are written in Puppet are first converted to a compiled format called catalog and later those catalogs are applied on the target machine.

#### **Visualpath Training & Consulting.**

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

## Puppet Agent

Puppet Agents are the actual working machines which are managed by the Puppet master. They have the Puppet agent daemon service running inside them.

### The Work Flow:

The first thing that Puppet master does is to collect the details of The Puppet agent. Using the factor which is present on all Puppet agents it gets all the machine level configuration details. All these details are gathered and sent back to the Puppet master.

Then the puppet master compares the gathered configuration with defined configuration details, and with the defined configuration it creates a catalog and sends it to the targeted Puppet nodes.

The Puppet node applies those configurations to get the system into the desired state.

After puppet agent changing to desired state, that node sends a report back to the Puppet master. This helps the Puppet master in understanding where the current state of the system is, as defined in the catalog.

## 2. INSTALLING PUPPET

### Prerequisites

Before we get started with installing Puppet, ensure that you have the following prerequisites:

- **Private Network DNS:** Forward and reverse DNS must be configured, and every server must have a unique hostname. If you do not have DNS configured, you must use your hosts file for name resolution.
- **Firewall Open Ports:** The Puppet master must be reachable on port 8140.
- **Install NTP:** Because it acts as a certificate authority for agent nodes, the puppet master server must maintain accurate system time to avoid potential problems when it issues agent certificates--certificates can appear to be expired if there are time discrepancies. We will use Network Time Protocol (NTP) for this purpose.

Install it with the following apt command

```
sudo apt-get update && sudo apt-get -y install ntp
```

### Install Puppet Master

Download the Puppet Labs package, with the following command

NOTE: For the time being we have taken centos

```
#yum install -y http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
```

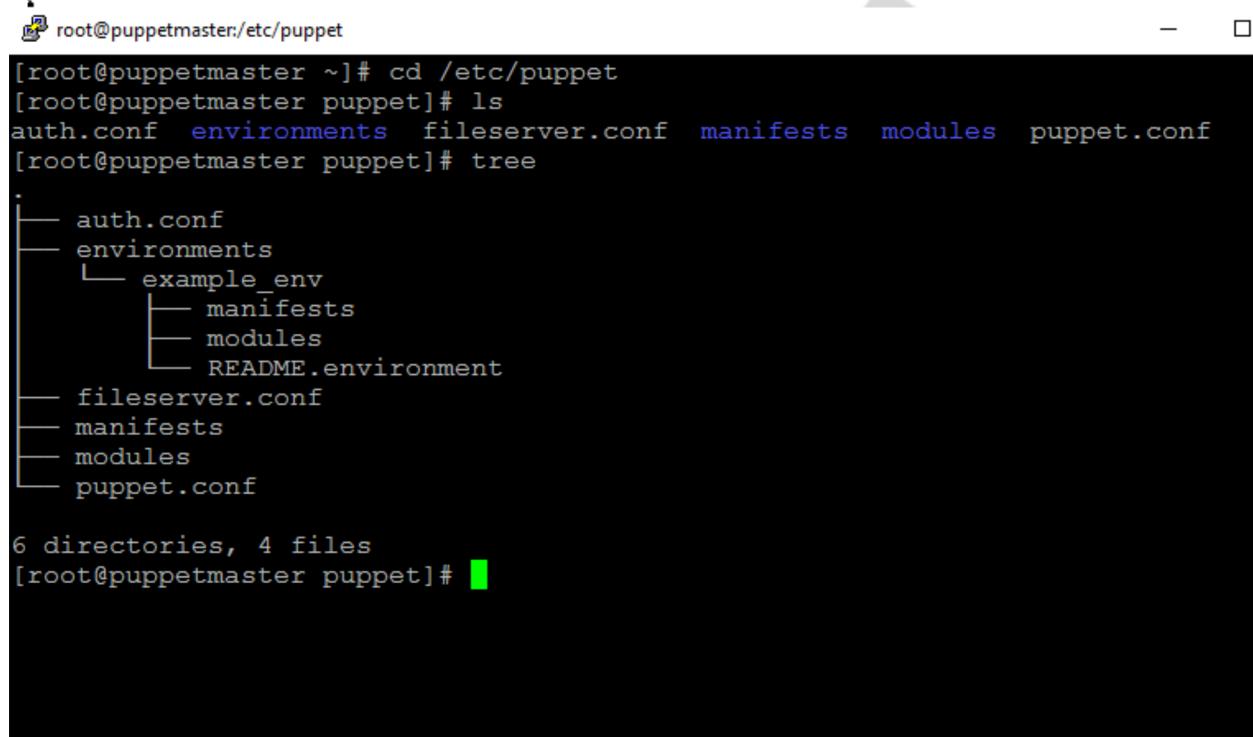
### Visualpath Training & Consulting

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

## Install the package

```
#yum install -y puppet-server
```

After installing Puppet server, we can check the file structure as seen below.



```
root@puppetmaster:~]# cd /etc/puppet
[root@puppetmaster puppet]# ls
auth.conf environments fileserver.conf manifests modules puppet.conf
[root@puppetmaster puppet]# tree
.
├── auth.conf
├── environments
│   └── example_env
│       ├── manifests
│       └── modules
│           └── README.environment
├── fileserver.conf
└── manifests
└── modules
└── puppet.conf

6 directories, 4 files
[root@puppetmaster puppet]#
```

## Install Puppet Agent:

```
#yum install -y http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
```

```
#yum install -y puppet
```

## Memory Allocation

By default, Puppet Server will be configured to use 2GB of RAM. However, if you want to experiment with Puppet Server on a VM, you can safely allocate as little as 512MB of memory. To change the Puppet Server memory allocation:

Open /etc/sysconfig/puppetserver and modify these settings:

```
# Modify this if you'd like to change the memory allocation, enable JMX, etc
JAVA_ARGS="-Xms2g -Xmx2g"
```

Replace 2g with the amount of memory you want to allocate to Puppet Server. For example, to allocate 1GB of memory, use JAVA\_ARGS="-Xms1g -Xmx1g"; for 512MB, use JAVA\_ARGS="-Xms512m -Xmx512m".

For more information about the recommended settings for the JVM, please see [Oracle's docs on JVM tuning](#).

Restart the puppetserver service after making any changes to this file.

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).