

It is important to note the use of the args section in the task. This is an optional way of listing the module options, and in this case, removes any confusion between what is a module option and what is the command itself.

### *Final apache.yml Playbook*

Now let's apply these changes. Open apache.yml.

```
$ vi apache.yml
```

With all of the changes above, change your apache.yml playbook to look like this.

```
---
```

```
- hosts: apache
```

```
  sudo: yes
```

```
  vars:
```

```
    http_port: 80
```

```
    domain: keplar.com
```

```
  tasks:
```

```
    - name: install apache2
```

```
      apt: name=apache2 update_cache=yes state=latest
```

```
    - name: enabled mod_rewrite
```

```
      apache2_module: name=rewrite state=present
```

```
      notify:
```

```
        - restart apache2
```

```
    - name: apache2 listen on port {{ http_port }}
```

```
      lineinfile: dest=/etc/apache2/ports.conf regexp="^Listen " line="Listen {{ http_port }}" state=present
```

```
      notify:
```

```
        - restart apache2
```

```
    - name: apache2 virtualhost on port {{ http_port }}
```

```
      lineinfile: dest=/etc/apache2/sites-available/000-default.conf regexp="^<VirtualHost \*:{{ http_port }}>" line="{{ <VirtualHost *:{{ http_port }}> }}
```

```
      notify:
```

```

- restart apache2

- name: create virtual host file
  template: src=virtualhost.conf dest=/etc/apache2/sites-available/{{ domain }}.conf

  -
  - name: a2ensite {{ domain }}
    command: a2ensite {{ domain }}
    args:
      creates: /etc/apache2/sites-enabled/{{ domain }}.conf
    notify:
      - restart apache2

  handlers:
    - name: restart apache2
      service: name=apache2 state=restarted

```

Save and close the file, then run the playbook.

```
$ ansible-playbook apache.yml --ask-sudo-pass
```

If you now visit the hostname or IP address of your secondary server in your browser, you will see it responds on port 80 again, not port 9090. Next, visit the domain (i.e. keplar.com) we specified for the new virtual host. Because we haven't added any files in yet, it should show an Apache 404 error page rather than the Apache welcome page. If so, your virtual host is working correctly, and you still haven't SSH'ed into your secondary server to run a single command.

## 30. Multi-Tier Web Application Stack Deployment using Ansible.

This project is about setting up & managing Loadbalancer, application server and database server with Ansible.

In this project, we will use as many as features of ansible to understand Ansible more in depth.

### **Prerequisites**

- Create 5 vm's with vagrant/ubuntu/trusty64 box.
- Control [Ansible server]
- lb01

### **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).

- app01
- app02
- db01
  
- SSH key exchange should be done from control to all 4 vm's.
- All vagrant vm's should have private static IP.
- Mention host to IP mapping in /etc/hosts file in all the vm's. As shown below.

  - 192.168.1.5 control
  - 192.168.1.6 lb01
  - 192.168.1.7 app01
  - 192.168.1.8 app02
  - 192.168.1.9 db01

Create directories on your local machines

```
# mkdir ansible_proj
# mkdir ansible_proj/Vms
# mkdir ansible_proj/control_repo
```

Place the vagrant file in Vms directory and Spin vagrant vm's.

```
# cd learn_proj/Vms
# vagrant up
```

### **Creating Python web application for deployment.**

Please refer below the content of the python web application and create it.

```
mkdir ansible_proj/control_repo/visualapp
mkdir ansible_proj/control_repo/visualapp/app
```

### **cat ansible\_proj/control\_repo/visualapp/visualapp.conf**

```
<VirtualHost *>
    WSGIDaemonProcess visualapp threads=5
    WSGIScriptAlias / /var/www/visualapp/visualapp.wsgi

    <Directory /var/www/visualapp>
        WSGIProcessGroup visualapp
        WSGIApplicationGroup %{GLOBAL}
        Order deny,allow
        Allow from all
    </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).

```
</VirtualHost>
```

### **cat ansible\_proj/control\_repo/visualapp/app/requirements.txt**

```
Flask==0.10.1
```

```
Flask-SQLAlchemy==2.0
```

### **cat ansible\_proj/control\_repo/visualapp/app/visualapp.py**

```
from flask import Flask

from flask.ext.sqlalchemy import SQLAlchemy

import os, socket


app = Flask(__name__)

app.config['SQLALCHEMY_DATABASE_URI'] = os.environ['DATABASE_URI']

db = SQLAlchemy(app)

hostname = socket.gethostname()

@app.route('/')

def index():

    return 'Hello, from VisualPath IT Academy %s!\n' % hostname

@app.route('/db')

def dbtest():

    try:

        db.create_all()

    except Exception as e:

        return e.message + '\n'

    return 'Database Connected from %s!\n' % hostname

if __name__ == '__main__':

    app.run()
```

### **cat ansible\_proj/control\_repo/visualapp/app/visualapp.wsgi**

```
activate_this = '/var/www/visualapp/.venv/bin/activate_this.py'
```

```
execfile(activate_this, dict(__file__=activate_this))
```

### **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).

```
import os  
os.environ['DATABASE_URI'] = 'mysql://visualapp:visualapp@db01/visualapp'  
  
import sys  
sys.path.insert(0, '/var/www/visualapp')  
  
from visualapp import app as application
```

## Ansible installation

Ansible package needs to be installed on control machine

Refer ansible doc

[http://docs.ansible.com/ansible/intro\\_installation.html#latest-releases-via-apt-ubuntu](http://docs.ansible.com/ansible/intro_installation.html#latest-releases-via-apt-ubuntu)

## Verify installation

```
# ansible --help  
# ansible-playbook --help  
# ansible-galaxy --help
```

## Setting up ansible repo

Login to control server

```
# vagrant ssh control  
# mkdir -p /home/vagrant/repo/ansible  
# cd /home/vagrant/repo/ansible  
# vi dev (inventory)  
+[loadbalancer]  
+lb01  
  
+[webserver]  
+app01  
+app02  
  
+[database]  
+db01
```

```
+[control]
```

## 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).

```
+control ansible_connection=local
```

## **Ansible.cfg**

Set inventory path in ansible.cfg

Benefit of tracking it in version control system

```
# cd /home/vagrant/repo/ansible  
# vi ansible.cfg
```

```
+[defaults]  
+inventory = ./dev
```

```
# ansible --list-hosts all
```

Validate the inventory and connections

```
# ansible --list-hosts all
```

Can use pattern syntax that allow you to select subset from the inventory.

```
# ansible --list-hosts all  
# ansible --list-hosts **  
# ansible --list-hosts loadbalancer  
# ansible --list-hosts webserver  
# ansible --list-hosts db01  
# ansible --list-hosts database:control  
# ansible --list-hosts webserver[0]  
# ansible --list-hosts \!control
```

[https://docs.ansible.com/ansible/intro\\_patterns.html](https://docs.ansible.com/ansible/intro_patterns.html)

[https://docs.ansible.com/ansible/intro\\_getting\\_started.html#your-first-commands](https://docs.ansible.com/ansible/intro_getting_started.html#your-first-commands)

[https://docs.ansible.com/ansible/ping\\_module.html](https://docs.ansible.com/ansible/ping_module.html)

[https://docs.ansible.com/ansible/command\\_module.html](https://docs.ansible.com/ansible/command_module.html)

```
# ansible -m ping all  
# ansible -m command -a "hostname" all
```

## **Default module is command**

```
# ansible -a "hostname" all
```

All tasks are going to have return status

Return exit code non-zero  

```
# ansible -a "/bin/false" all
```

[http://docs.ansible.com/ansible/modules\\_by\\_category.html](http://docs.ansible.com/ansible/modules_by_category.html)

Plays

## **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 /home/vagrant/repo/ansible
# vi hostname.yml
+ ---
+ - hosts: all
+   tasks:
+     - command: hostname

` Playbook Execution
# ansible-playbook hostname.yml
# vi hostname.yml
---
- hosts: all
  tasks:
+   - name : get server info
-   - command: hostname
+   command: hostname

# ansible-playbook hostname.yml

```

## **Four pillars of Linux application | Principles to setup app on Linux**

1. Packages': From repositories (apt or yum) or any other resources
2. Services: init.d or system.d or your own start script
3. System configurations: Files, directories, users, permission, firewall rules etc
4. Config files for the app itself

### Playbook intro

[http://docs.ansible.com/ansible/modules\\_by\\_category.html](http://docs.ansible.com/ansible/modules_by_category.html)

### Packages

Creating playbook for installing nginx package on loadbalancer

```

# vi loadbalancer.yml
---

- hosts: loadbalancer
become: true
tasks:
  - name: Install nginx
    apt: name=nginx state=present update_cache=yes

```

Creating playbook for installing mysql-server package on database server

```
# vi database.yml
```

```
---
- hosts: database
```

### **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).

```
become: true
tasks:
  - name: Install mysql-server
    apt: name=mysql-server state=present update_cache=yes
```

Execute loadbalancer playbook

```
#ansible-playbook loadbalancer.yml
Execute it again
# ansible-playbook loadbalancer.yml
```

Execute Database playbook

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

Creating webserver playbook, which covers loops and jinja2 templates  
[https://docs.ansible.com/ansible/playbooks\\_loops.html#standard-loops](https://docs.ansible.com/ansible/playbooks_loops.html#standard-loops)  
<http://jinja.pocoo.org/>

```
# vi webserver.yml
```

```
---
- hosts: webserver
  become: true
  tasks:
    - name: Install apache2
      apt: name={{item}} state=present update_cache=yes
      with_items:
        - apache2
        - libapache2-mod-wsgi
        - python-pip
        - python-virtualenv
```

Services modules

```
# vi loadbalancer.yml
+
+   - name: Ensure nginx started
+     service: name=nginx state=started enabled=yes
```

Test the nginx service

```
# wget -qO- http://lb01 | less
```

Easier to do with curl, so install curl on the control server with playbook

```
# vi control.yml
```

**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).

```
---
- hosts: control
  become: true
  tasks:
    - name: install tools
      apt: name={{item}} state=present update_cache=yes
      with_items:
        - curl
```

```
# ansible-playbook control.yml
# curl lb01
```

Add service module to webserver and database playbooks  
# vi webserver.yml

```
+ - name: Ensure Apache started
+   service: name=apache2 state=started enabled=yes
```

```
# vi database.yml
+ - name: Ensure Mysql started
+   service: name=mysql state=started enabled=yes
```

```
# $ ansible-playbook webserver.yml
```

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

### apache2\_module, handlers, notify

We need to setup the apache receiving python application. Python app going to use mod-wsgi to serve the request. Make sure mod-wsgi is enabled with apache\_module.

[https://docs.ansible.com/ansible/apache2\\_module.html](https://docs.ansible.com/ansible/apache2_module.html)

[https://docs.ansible.com/ansible/playbooks\\_intro.html#handlers-running-operations-on-change](https://docs.ansible.com/ansible/playbooks_intro.html#handlers-running-operations-on-change)

```
# ansible-playbook webserver.yml
```

```
+ - name: Ensure mod_wsgi enables
+   apache2_module: state=present name=wsgi
+   notify: Restart apache2
+ handlers:
+   - name: Restart apache2
+     service: name=apache2 state=restarted
```

```
# ansible-playbook webserver.yml
```

## Files: Copy

Copy the visualapp directory in the ansible directory which contains the python app written in flask.  
Use copy module to ship application folder from control server to app01/02 servers.

Our visualapp site file also needs to be copied in /etc/apache2/sites-available directory.

We will enable our visualapp website in next section.

```
# vi webserver.yml
```

```
+ - name: copy visualapp app source
+   copy: src=visualapp/app/ dest=/var/www/visualapp mode=0755
+   notify: Restart apache

+ - name: copy apache virtual host config
+   copy: src=visualapp/visualapp.conf dest=/etc/apache2/sites-available mode=0755
+   notify: Restart apache
```

```
# ansible-playbook webserver.yml
```

```
# curl app01
```

Still shows the default apache site for python app working we need to configure PIP and Virtualenv

## Application Modules

To run the python flask application, we need to install flask & SQLAlchemy python package.

To install python packages, we will use 'pip' which is a python package manager.

But we will not install the python packages directly on the system, we will create a virtual environment/container known as virtualenv which will hold all our python packages.

```
# vi webserver.yml
```

```
+ - name: setup python virtualenv
+   pip: requirements=/var/www/visualapp/requirements.txt virtualenv=/var/www/visualapp/.venv
+   notify: Restart apache2
```

```
# ansible-playbook webserver.yml
```

## Files: File | Activating python site and deactivating apache default site

By default, apache will serve a website, also known as apache default page.

We will disable the default website and enable our visualapp website.

To disable default website, we need to unlink 000-default.conf file located in sites-enabled directory. Next, we will create a link from /etc/apache2/sites-available/visualapp.conf to /etc/apache2/sites-enabled/visualapp.conf.

```
# vi webserver.yml
```

```
+ - name: de-activate default apache site
+   file: path=/etc/apache2/sites-enabled/000-default.conf state=absent
+   notify: Restart apache
```

## 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: activate visualapp apache site
+   file: src=/etc/apache2/sites-available/visualapp.conf dest=/etc/apache2/sites-
enabled/visualapp.conf state=link
+   notify: Restart apache

# ansible-playbook webserver.yml
# curl app01
# curl app02
# curl lb01, will show the default page from nginx, we need to configure it to point to the App
servers

```

### **Templates | Configure lb01 to point to app01 & app02**

By default, nginx will serve a website, also known as nginx default page.

We will disable the default website and enable a redirect rule which will forward request from nginx to our app servers randomly.

To disable default website, we need to unlink default file located in /etc/nginx/sites-enabled/ directory.

Next, we will create a template templates/nginx.conf.j2 as mentioned below and push that template to /etc/nginx/sites-available/visualapp location on lb01.

Once we push visualapp configuration file in site-available directory we can create a link from /etc/nginx/sites-enabled/visualapp to /etc/nginx/sites-available/visualapp which will enable the nginx site.

Nginx site is a redirect rule which will forward the requests to app01 & app02 randomly

```

# curl lb01
# mkdir templates
# vi templates/nginx.conf.j2
+upstream visualapp {
+{% for server in groups.webserver %}
+  server {{ server }};
+{% endfor %}
+}

+server {
+  listen 80;

+  location / {
+    proxy_pass http://visualapp;
+  }
+}

# vi loadbalancer.yml

+ - name: configure nginx site
+   template: src=templates/nginx.conf.j2 dest=/etc/nginx/sites-available/visualapp mode=0644
+   notify: restart nginx

+ - name: de-activate default nginx site

```

### **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).

```

+   file: path=/etc/nginx/sites-enabled/default state=absent
+   notify: restart nginx

+   - name: activate visualapp nginx site
+     file: src=/etc/nginx/sites-available/visualapp dest=/etc/nginx/sites-enabled/visualapp state=link
+     notify: restart nginx

+ handlers:
+   - name: restart nginx
+     service: name=nginx state=restarted

# ansible-playbook loadbalancer.yml

```

Test if it's working.

```
# curl lb01
```

### **Lineinfile | Make db server to listen to on all interface 0.0.0.0.**

[https://docs.ansible.com/ansible/lineinfile\\_module.html](https://docs.ansible.com/ansible/lineinfile_module.html)

Test the connection first

```
# curl app01/db
```

- Check visualapp.wsgi which has db connection info.
- Check visualapp.py which establishes connection from app to db server
- Login to db server and see mysql listening only on local interface

```
# netstat -an
```

- Open /etc/mysql/my.cnf

With ansible lineinfile module replace “bind-address 127.0.0.1” to bind-address to “bind-address 0.0.0.0”

Login to control server

```
# vi database.yml
```

```

+   - name: ensure mysql listening on all ports
+     lineinfile: dest=/etc/mysql/my.cnf regexp='^bind-address'
+       line="bind-address = 0.0.0.0"
+
+     notify: restart mysql

+ handlers:
+   - name: restart mysql
+     service: name=mysql state=restarted

```

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

### **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).

**Application Modules: mysql\_db, mysql\_user | Install python-mysqldb in app and db server, create visualapp db and visualapp user**

Install mysql module in apache server

```
# vi webserver.yml
- name: Install apache2
  apt: name={{item}} state=present update_cache=yes
  with_items:
    - apache2
    - libapache2-mod-wsgi
    - python-pip
    - python-virtualenv
+   - python-mysqldb
```

```
# ansible-playbook webserver.yml
```

[https://docs.ansible.com/ansible/mysql\\_db\\_module.html](https://docs.ansible.com/ansible/mysql_db_module.html)  
[https://docs.ansible.com/ansible/mysql\\_user\\_module.html](https://docs.ansible.com/ansible/mysql_user_module.html)

```
# vi database.yml
```

```
- hosts: database
  become: true
  tasks:
+   - name: install tools
+     apt: name={{item}} state=present update_cache=yes
+     with_items:
+       - python-mysqldb
+
- name: install mysql-server
  apt: name=mysql-server state=present update_cache=yes
```

```
lineinfile: dest=/etc/mysql/my.cnf regexp='^bind-address' line="bind-address = 0.0.0.0"
notify: restart mysql
```

```
+   - name: create visualapp database
+     mysql_db: name=visualapp state=present
+
+   - name: create visualapp user
```

```
+   mysql_user: name=visualapp password=visualapp priv=visualapp.*:ALL host='%'  
state=present  
+  
handlers:  
- name: restart mysql  
  service: name=mysql state=restarted
```

Execute db playbook  
# ansible-playbook database.yml

Test the connection

```
# curl app01/db  
# curl app02/db  
# curl lb01/db  
# curl lb01/db  
# curl lb01/db  
# curl lb01/db
```

### Roles:

We have three tiers in our app, for every tier we have a separate playbook.  
If we have another team, they have their application which they want to manage with ansible.  
How much of our existing code is reusable to our new team?  
We can copy all our playbooks paste them and go through each line and edit every place we put visualapp and replace that with new application detail.  
That is very tedious and especially we have more application coming down the line.  
Also, if they have different config like they want to use different ports.  
If we want to make changes to our database configuration, we got many places in all our playbooks that require changes. Lot of code duplication there.

Ansible gives us Roles to solve such problems. Essentially, we need to create a folder for every role which will have sub folders like tasks, files, handlers, vars etc, which holds all the information separately.

For example, all the tasks that we have written in our playbooks goes into tasks folder. Now instead of having all the configuration for our tiers mashed up in one playbook we can distribute it into multiple folders or files.

Encapsulation

Code Reusability

Scalability

Ansible galaxy roles

```
# mkdir -p /home/vagrant/repo/ansible-roles  
# cp -r /home/vagrant/repo/ansible/* /home/vagrant/repo/ansible-roles/  
# cd /home/vagrant/repo/ansible-roles  
# mkdir roles  
# cd roles
```

### 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).

Create skeleton of roles with ansible-galaxy

```
# ansible-galaxy init control
# ansible-galaxy init mysql
# ansible-galaxy init nginx
# ansible-galaxy init apache2
# ansible-galaxy init visualapp_app
```

Need to move content of all the playbooks to their respective roles

### 1. **control.yml:**

Move lists of tasks to control role's tasks/main.yml  
Replace tasks section from the control playbook with roles content.  
# vi control.yml

```
---
- hosts: control
  become: true
  roles:
    - control
```

Execute control.yml to test.

```
# ansible-playbook control.yml
```

### 2. **database.yml:**

Move lists of tasks to mysql role's tasks/main.yml.  
Move mysql start task after my.cnf file change section.  
Move lists of handlers to mysql role's handlers/main.yml.  
Replace tasks & handlers section from the playbook with roles content.

```
---
- hosts: database
  become: true
  roles:
    - mysql
```

### 3. **loadbalancer.yml**

Move lists of tasks to nginx role's tasks/main.yml.  
Move lists of handlers to nginx role's handlers/main.yml.  
Move nginx.conf.j2 template to nginx role's templates directory  
Open tasks/main.yml of the nginx role, change templates src path  
from src= templates/nginx.conf.j2 to src=nginx.conf.j2  
Replace tasks & handlers section from the playbook with roles content.

```
---
- hosts: loadbalancer
  become: true
  roles:
    - nginx
```

#### 4. webserver.yml

Move lists of apache tasks to apache role's tasks/main.yml.  
Move apache2 start task to the end of the.  
Move lists of apache handlers to apache role's handlers/main.yml.  
Move lists of app tasks to visualapp\_app role's tasks/main.yml.  
Move lists of app handlers to visualapp\_app role's handlers/main.yml.  
Copy visualapp app directory to visualapp\_app role's files/

Replace tasks & handlers section from the playbook with roles content.

```
---  
- hosts: webserver  
become: true  
roles:  
- apache2  
- visualapp_app
```

#### Site.yml: include | include all the playbook into site.yml

Create site.yml at the same place where all the playbooks are located

```
# vi site.yml
```

```
---  
- include: control.yml  
- include: database.yml  
- include: webserver.yml  
- include: loadbalancer.yml
```

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

#### Variables: facts | Will use fact variable to make mysql listen only on its own IP address and not on 0.0.0.0

Get the list of all the fact variable with setup module

```
# ansible -m setup database
```

We will use ansible\_eth0.ipv4.address fact variable in our lineinfile module for the bind-address instead of 0.0.0.0

Edit database role's tasks

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

```
- name: ensure mysql listening on all ports  
- lineinfile: dest=/etc/mysql/my.cnf regexp='^bind-address' line="bind-address = 0.0.0.0"
```

#### 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).

```
+ lineinfile: dest=/etc/mysql/my.cnf regexp='^bind-address'
+     line="bind-address = {{ ansible_eth0.ipv4.address }}"
+     notify: restart mysql

- name: ensure mysql started
```

Execute stack\_status.yml  
#ansible-playbook stack\_status.yml

We also need to update the wait\_for module in stack\_status playbook.  
By default, it will wait for the loopback address 127.0.0.1. Since we updated mysql config to only listen on its ipaddress we need to make change in stack\_status playbook as well.

Update stack\_status.yml  
# vi playbooks/stack\_status.yml

```
- name: verify mysql is listening on 3306
-   wait_for: port=3306 timeout=1
+   wait_for: host={{ ansible_eth1.ipv4.address }} port=3306 timeout=1
```

Also update wait\_for module in stack\_restart playbook.

Update stack\_restart.yml  
# vi playbooks/stack\_restart.yml

```
become: true
tasks:
  - service: name=mysql state=restarted
  - - wait_for: port=3306 state=started
+  - wait_for: host={{ ansible_eth1.ipv4.address }} port=3306 state=started
```

### Variables: defaults | Using variable in mysql\_db & mysql\_user module

We have hard coded db config parameter like user, password, dbname etc in the playbook which is not modular, if we want to change a parameter then we need to go over the entire playbook and change every place where we mentioned that parameter like db name.

### 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).

We will use Role's defaults directory and put all our db config info into variables. We can then refer to these variables in our playbook.

Lowest priorities are for default vars, refer the doc.

[https://docs.ansible.com/ansible/playbooks\\_variables.html#variable-precedence-where-should-i-put-a-variable](https://docs.ansible.com/ansible/playbooks_variables.html#variable-precedence-where-should-i-put-a-variable)

roles/mysql/defaults/main.yml

```
---
-# defaults file for mysql
+db_name: myapp
+db_user_name: dbuser
+db_user_pass: dbpass
+db_user_host: localhost
```

roles/mysql/tasks/main.yml

```
- name: ensure mysql started
  service: name=mysql state=started enabled=yes

-- name: create visualapp database
- mysql_db: name=visualapp state=present
++ name: create database
+ mysql_db: name={{ db_name }} state=present

-- name: create visualapp user
- mysql_user: name=visualapp password=visualapp priv=visualapp.*:ALL host='%' state=present
+- name: create user
+ mysql_user: name={{ db_user_name }} password={{ db_user_pass }} priv={{ db_name }}.*:ALL
+   host='{{ db_user_host }}' state=present
```

### Variables: Vars | Multiple ways/places to define vars, refer the doc.

[https://docs.ansible.com/ansible/playbooks\\_variables.html#variable-precedence-where-should-i-put-a-variable](https://docs.ansible.com/ansible/playbooks_variables.html#variable-precedence-where-should-i-put-a-variable)

Already defined vars in mysql role's defaults/main.yml which has the last precedence and would be used if vars are nowhere defined. We can override these variables by defining them while referring to the role in our database.yml playbook

We will define vars in database.yml playbook with a hash.

In this case variable defined in below playbook will have higher precedence than our default variable defined in above section.

```
# vi database.yml
- hosts: database
  become: true
  roles:
    - mysql
```

### 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).

```
+ - { role: mysql, db_name: visualapp, db_user_name: visualapp, db_user_pass: visualapp,  
db_user_host: '%' }
```

Variables: with\_dict | Use of with\_dict module in nginx.yml playbook.

We can also provide variables to playbooks in list format as mentioned below.

Will provide lists of variables in nginx role's defaults/main.yml. As specified below for reference.

Name of the list is "sites".

```
---
```

```
# defaults file for nginx
```

```
sites:
```

```
    myapp:
```

```
        frontend: 80
```

```
        backend: 80
```

Will refer to those variables with the module with\_dict in nginx.yml playbook. As specified below for reference.

with\_dict: sites

with\_dict will supply variable name in hash/dictionary format, variable names are {{ item.key }} & {{ item.value.frontend }} & {{ item.value.backend }}

```
(item={'value': {u'frontend': 80, u'backend': 80}, 'key': u'myapp'})
```

Update nginx role's defaults/main.yml

```
+sites:  
+ myapp:  
+   frontend: 80  
+   backend: 80
```

Update nginx role's tasks/main.yml

```
- name: install nginx  
  apt: name=nginx state=present update_cache=yes  
  
-- name: configure nginx site  
- template: src=nginx.conf.j2 dest=/etc/nginx/sites-available/visualapp mode=0644  
+- name: configure nginx sites  
+ template: src=nginx.conf.j2 dest=/etc/nginx/sites-available/{{ item.key }} mode=0644  
+ with_dict: sites  
  notify: restart nginx  
  
- name: de-activate default nginx site  
  file: path=/etc/nginx/sites-enabled/default state=absent  
  notify: restart nginx  
  
-- name: activate visualapp nginx site  
- file: src=/etc/nginx/sites-available/visualapp dest=/etc/nginx/sites-enabled/visualapp state=link  
+- name: activate nginx sites
```

## 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).

```
+ file: src=/etc/nginx/sites-available/{{ item.key }} dest=/etc/nginx/sites-enabled/{{ item.key }}
state=link
+ with_dict: sites
  notify: restart nginx
```

Update nginx role's templates/nginx.conf.j2

```
-upstream visualapp {
+upstream {{ item.key }} {
  {% for server in groups.webserver %}
-   server {{ server }};
+   server {{ server }}:{{ item.value.backend }};
  {% endfor %}
}

server {
-   listen 80;
+   listen {{ item.value.frontend }};

  location / {
-     proxy_pass http://visualapp;
+     proxy_pass http://{{ item.key }};
  }
}
```

### **Variables continued | will push visualapp.wsgi as a template from visualapp\_app role**

We have replaced hardcoded DB values in our playbook as variables.

We also have db config parameters hardcoded in visualapp.wsgi file from our python flask app.  
We will push visualapp.wsgi as a template and put all our db variables into that template.

Move visualapp.wsgi to visualapp\_app role's templates directory

```
# mv roles/visualapp_app/files/visualapp/app/visualapp.wsgi
roles/visualapp_app/templates/visualapp.wsgi.j2
```

Update the visualapp.wsgi.j2 with db variables names.

```
activate_this = '/var/www/visualapp/.venv/bin/activate_this.py'
execfile(activate_this, dict(__file__=activate_this))

import os
+os.environ['DATABASE_URI'] = 'mysql://{{ db_user }}:{{ db_pass }}@{{ groups.database[0] }}/{{ db_name }}'

import sys
sys.path.insert(0, '/var/www/visualapp')
```

### **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).