POC3 : Nginx as proxy server Through Ansible

Step 1: Create Directory Structure as follows-

-proxyServer

|-----proxyServer\_role

|----handlers

|----tasks

|----templates

|----vars

|----files

|----defaults

Step 2: In “tasks” folder create “install.yml” file and write following content in it.

For installing LNMP we need to install Apache, PHP, Mysql and this playbook does the same.

|  |
| --- |
| *---*  *- name: 1. install Nginx*  *apt: name=nginx state=present*  *- name: 2. install PHP and PHP module for Nginx*  *apt: name={{item}} state=present*  *with\_items:*   * *php5* * *php5-cli* * *php5-cgi* * *psmisc* * *spawn-fcgi* * *php5-fpm* * *php5-gd* * *php-xml-parser* * *php5-intl* * *php5-curl* * *php5-mysql*   *notify:*  *- restart nginx*  *- reload nginx*  *- name: MySQL | Make sure the MySql packages are installed*  *apt: name={{items}} state=present*  *with\_items:*   * *mysql-server* * *mysql-client* * *python-mysqldb*   *- name: MySQL | Ensure MySQL is running*  *service: name=mysql state= started* |

Step 3: In “defaults” folder create “main.yml” file and write following content in it.

This file consist of some default values for some variables. We have given file name as main.yml so that when we call role, the content of this file should automatically get loaded.

|  |
| --- |
| *---*  # Basic settings  mysql\_root\_password: 'password'  mysql\_language: '/usr/share/mysql/'  # List of databases to be created  mysql\_databases: [demo]  # List of users to be created  mysql\_users: [amol] |

Step 4: In “tasks” folder create “users.yml” file and write following content in it.

This file manages users for MySql. It creates users mentioned in mysql\_users array

|  |
| --- |
| *---*  *- name: MySQL | Make sure the MySQL users are present*  *mysql\_user: name={{item}} password=password priv=\*.\*:ALL state=present host=localhost*  *with\_items: mysql\_users*  *when: mysql\_users|length > 0* |

Step 5: In “tasks” folder create “databases.yml” file and write following content in it.

This file manages databases for MySql. It creates databases mentioned in mysql\_databases array

|  |
| --- |
| *---*  *- name: MySQL | Make sure the MySQL databases are present*  *mysql\_db:*  *name: "{{ item.name }}"*  *state: present*  *with\_items: mysql\_databases*  *when: mysql\_databases|length > 0*  - name: MySQL | copy sample database  copy: src=dump.sql dest=/tmp/dump.sql    - name: MySQL | insert sample data  mysql\_db: name=demo state=import target=/tmp/dump.sql |

Step 6: In “tasks” folder create “secure.yml” file and write following content in it.

This file sets password for root user of MySql and configure it for easy access, removes anonymous users and test database and execute sql script to dump data in particular database.

|  |
| --- |
| *---*  *- name: MySQL | Configure MySql for easy access as root user*  *template:*  *src: root\_dot\_my.cnf.j2*  *dest: /home/amol/.my.cnf*  *owner: amol*  *group: amol*  *mode: 0600*  *- name: MySQL | Remove anonymous MySQL server user*  *mysql\_user:*  *name: ""*  *host: "{{item}}"*  *state: absent*  *with\_items:*  *- localhost*  *- name: MySQL | Remove the MySQL test database*  *mysql\_db:*  *name: test*  *state: absent*  - name: MySQL | copy sample database  copy: src=dump.sql dest=/tmp/dump.sql    - name: MySQL | insert sample data  shell: cat /tmp/dump.sql | mysql -u amol -pamol password |

Step 7: In “tasks” folder create “configure.yml” file and write following content in it.

|  |
| --- |
| *---*  *- name: Copy the nginx default site configuration file*  *template: src=default.j2 dest=/etc/nginx/sites-available/default*    *- name: rename nginx default site configuration file to desired webapp*  *command: /bin/mv /etc/nginx/sites-available/default /etc/nginx/sites-available/project\_name*  *- name: remove symlink for default*  *command: /bin/rm /etc/nginx/sites-available/default*  *- name: create new symlink*  *file: src=/etc/nginx/sites-available/project\_name dest=/etc/nginx/sites-enabled/project\_name owner=amol group=amol state=link*  *notify:*  *- restart nginx*  *- reload nginx*  *- name: create cache folder for caching static content*  *command: /bin/mkdir /var/www/html/cache* |

Step 8: In “tasks” folder create “deploy.yml” file and write following content in it.

This file deploys your PHP project to the Document root of remote host

|  |
| --- |
| *---*  *- name: Deploy PHP project*  *copy: src=/var/www/html/project\_name dest=/var/www/html/* |

Step 9: In “tasks” folder create “main.yml” file and write following content in it.

|  |
| --- |
| *---*  *- include: install.yml*  *- include: secure.yml*  *- include: databases.yml*  *- include: users.yml*  *- include: configure.yml*  *- include: deploy.yml* |

Step 10: In “handlers” folder create “main.yml” file and write following content in it.

|  |
| --- |
| *---*  *- name: restart nginx*  *service: name=nginx state=restarted*  *- name: reload nginx*  *service: name=nginx state=reloaded* |

Step 11: In “templates” folder create “my\_cnf.j2” file and write following content in it.

|  |
| --- |
| *[client]*  *user=amol*  *password={{ mysql\_root\_password }}* |

Step 12: In “templates” folder create “default.j2” file and write following content in it.

|  |
| --- |
| *server {*  *listen 80;*  *server\_name {{server}};*  *root /var/www/html/website1;*  *index index.php index.html index.htm;*  *server\_name website1.com;*  *location / {*  *try\_files $uri $uri/ /index.php;*  *}*  *location ~ \.php$ {*    *proxy\_set\_header X-Real-IP $remote\_addr;*  *proxy\_set\_header X-Forwarded-For $remote\_addr;*  *proxy\_set\_header Host $host;*  *proxy\_pass http://{{proxy}};*  *}*  *location ~ /\.ht {*  *deny all;*  *}*  *# for storing static content*  *location ~\* \.(gif|jpg|jpeg|png|wmv|avi|mpg|mpeg|mp4|htm|html|js|css|mp3|swf|ico|flv)$ {*  *expires max;*  *proxy\_set\_header X-Real-IP $remote\_addr;*  *proxy\_pass http://{{proxy}};*  *proxy\_ignore\_headers X-Accel-Expires Expires Cache-Control;*  *proxy\_store /var/www/html/cache$uri;*  *proxy\_store\_access user:rw group:rw all:r;*  *}*    *}* |

Step 13: In “files” folder create “dump.sql” file and write following content in it.

|  |
| --- |
| *--*  *-- Database: `demo`*  *--*  *--*  *-- Table structure for table `user`*  *--*  *CREATE TABLE IF NOT EXISTS `user` (*  *`id` int(11) NOT NULL AUTO\_INCREMENT,*  *`name` varchar(300) NOT NULL,*  *`lastname` varchar(300) NOT NULL,*  *`department` varchar(300) NOT NULL,*  *`username` varchar(300) NOT NULL,*  *`password` varchar(300) NOT NULL,*  *PRIMARY KEY (`id`)*  *) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO\_INCREMENT=3 ;*  *--*  *-- Dumping data for table `user`*  *--*  *INSERT INTO `user` (`id`, `name`, `lastname`, `department`, `username`, `password`) VALUES*  *(1, 'argie', 'policarpio', 'OBD', 'admin', 'admin'),*  *(2, 'dsfdsds', 'sdsds', 'OGM', 'a', 'a');* |

Step 14: In “proxyServer” folder create “proxyServer.yml” file and write following content in it.

|  |
| --- |
| *---*  *- hosts: 172.27.59.17*  *Remote\_user: amol*  *sudo: yes*  *roles:*   * *proxyServer \_role* |

Step 15: Now to execute what we have written, execute following command

ansible-playbook proxyServer.yml - -ask-sudo-pass

Step 16: Edit /etc/hosts file and put

172.27.59.8 [www.project\_name.com](http://www.project_name.com)

Where 172.27.59.8 is ip of machine in which nginx installed.