Ansible IT Automation

Objective

- Understanding Ansible
- Set up Ansible on your workstation
- Write a playbook to install and configure Nginx
- Deploy a simple application

What is Ansible?

- It is a simple IT automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs.

- It is designed for multi-tier deployments.

- It models the IT infrastructure by describing how all of your systems inter-relate, rather than just managing one system at a time.

How does Ansible work?

- As it's written in python it requires python to be installed on the machine to run. It also needs jinja2, PyYAML.

- It uses SSH to communicate with the hosts and to execute tasks.

- The playbooks are written in YAML(Yet Another Markup Language) which is very human readable.

Why is Ansible different?

- It uses no agents and no additional custom security infrastructure.
- It’s very powerful.
- It’s very simple to use, deploy and light weight.

Setting up Ansible on Mac

- First install Xcode
- Next install pip (sudo easy_install pip)
- Install Ansible (sudo pip install ansible -quiet)
- Upgrade Ansible (sudo pip install ansible -upgrade)

Setting up Ansible on Linux

- First install python and git (apt-get install python-pip python-dev git -y)

- Second install PyYAML and jinja2 (pip install PyYAML jinja2 paramiko)

- Clone the Ansible repository (git clone https://github.com/ansible/ansible.git)

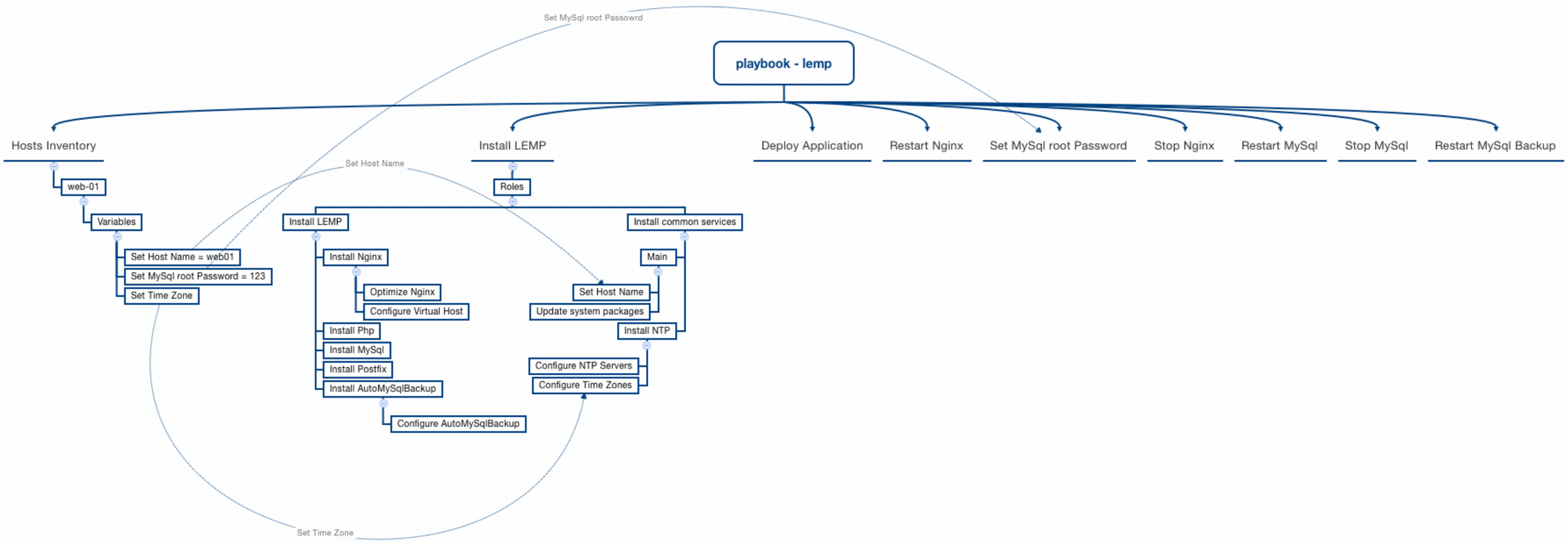
- Install Ansible (cd ansible && make install)

- Create global hosts (mkdir /etc/ansible && cp ~/ansible/examples/hosts /etc/ansible/)

Start writing a simple playbook

- Define the algorithm for the playbook
- Create an inventory
- Create roles and tasks

Algorithm


Set MySql root Passowrd
playbook - lemp
Hosts Inventory
Install LEMP
Deploy Application
Restart Nginx
Set MySql root Password
Stop Nginx
Restart MySql
Stop MySql
Restart MySql Backup
Set Host Name
web-01
Variables
Set Host Name = web01
Set MySql root Password = 123
Set Time Zone
Roles
Install LEMP
Install Nginx
Optimize Nginx
Configure Virtual Host
Install Php
Install MySql
Install Postfix
Install AutoMySqlBackup
Configure AutoMySqlBackup
Install common services
Main
Set Host Name
Update system packages
Install NTP
Configure NTP Servers
Configure Time Zones
Set Time Zone

Important Components of a playbook

- Inventory
- Variables
- Roles
- Tasks
- Handlers
- Templates

Inventory & Variables

- The inventory comprises of groups, host names hosts IP address and variables.

- Sample hosts entry:

```
[webserver]
web-01 ansible_ssh_host=192.168.44.135
ansible_ssh_user=adithya sethostname=web-01
setmysqlrootpassword=123 timezone=Asia/Calcutta
```

Roles & Tasks

- Roles can be used to segregate different types of server. e.g. Database/Web servers.

- Roles are further broken down into tasks, which help in reducing the complexity of the playbook.

Handlers

- Handlers are used to notify after a task is executed. e.g.

```
- name: restart ntp
service: name=ntp state=restarted
```

Templates

- Templates are written in jinja2 format.
- Sample template file

```
server {
      listen  80
        root /usr/share/nginx/html/phpapp
        index index.php index.html index.htm;
       server_name {{ sethostname }};
 location / {
              try_files $uri $uri/ /index.php?
$args;
       }
```

Executing a playbook

- A playbook can be executed using the ansible-ploybook command along with the relevant parameters.

- e.g.
```
ansible-playbook -i hosts install-lemp.yml
```

Understanding The Execution Output

- NOTIFIED: [lemp | start mysql]
  ****
  ok: [web-01]

- NOTIFIED: [lemp | restart nginx]
  ****
  changed: [web-01]

Questions ?

Thank you!