Ansible Network Configuration Automation Playbook

by Thanura Rukshan

Documentation

Overview

This documentation provides a detailed guide on the Network Configuration Automation playbook using Ansible. The playbook is designed to automate server configurations, including network setup, package management, system tuning, monitoring, disk partitioning, and security enhancements.

Prerequisites

Before running the playbook, ensure the following:

1. **Ansible Installation**: Install Ansible on the control node.

   ```
   sudo apt update && sudo apt install ansible -y  # Ubuntu/Debian

   sudo yum install ansible -y  # CentOS/RHEL
   ```

2. **SSH Access**: Ensure SSH access to target servers is configured.

   ```
   ssh-copy-id user@target-server
   ```

3. **Sudo Privileges**: The Ansible user should have sudo privileges.
4. **Inventory File**: Define the target servers in `playbooks/dev.ini`.
5. **Python and Dependencies**: Ensure Python and required packages (e.g., pip) are installed on the target servers.

   ```
   sudo apt install python3 python3-pip -y
   ```

Inventory File (`playbooks/dev.ini`)

Update this file by adding target server information.

```
[all]
testserver ansible_host=192.168.1.100 ansible_user=admin ansible_ssh_pass=your_password
```

Playbook Structure

File/Folder Structure

```
network-configuration-automation-ansible/
├── ansible.cfg
├── playbooks/
│   ├── site.yml
│   ├── dev.ini
├── roles/
│   ├── connectivity-test/
│   ├── network-configuration-set/
│   ├── user-management/
│   ├── package-management/
│   ├── system-tuning/
│   ├── monitoring/
│   ├── server-partitioning/
```

```
│   ├── grafana/
│   ├── node_exporter/
│   ├── prometheus_installation/
│   ├── snmp_notifier/
│   ├── chrony_config/
│   ├── selinux-configuration/
│   ├── systemctl_runlevel/
│   ├── yum_update_and_sync/
│   ├── scan/
```

The main playbook file is `playbooks/site.yml`, and it calls the following roles:

```yaml
- name: Test connectivity to target servers
  hosts: all
  gather_facts: no
  roles:
    - connectivity-test

- name: Configure Network
  hosts: all
  become: yes
  roles:
    - network-configuration-set
    - user-management
    - package-management
    - system-tuning
    - monitoring

- name: Disk Partitioning
  hosts: all
  become: yes
  roles:
    - server-partitioning

- name: Setup Monitoring
  hosts: all
  become: yes
  roles:
    - grafana
    - node_exporter
    - prometheus_installation
    - snmp_notifier

- name: OS Configurations
  hosts: all
  become: yes
  roles:
    - chrony_config
    - selinux-configuration
    - systemctl_runlevel
    - yum_update_and_sync

- name: Scan the server
  hosts: all
  become: yes
  roles:
    - scan
```

Role Descriptions

Each role consists of tasks, variables, and handlers.

1. connectivity-test

Purpose: Ensures Ansible can reach the target servers.

Tasks (`tasks/main.yml`):

```yaml
- name: Ping target servers
```

```
  ansible.builtin.ping:
```

2. network-configuration-set

Purpose: Configures network settings.

Variables (`vars/main.yml`):

```
---
# vars file for network-configuration-set
# roles/network-configuration-automation/vars/main.yml

# Network Interface Configuration
network_interface: eth0  # The network interface to configure (e.g., eth0, ens33)

# IP Configuration
use_dhcp: true  # Set to false for a static IP configuration
static_ip: "192.168.1.100"  # Static IP address (if use_dhcp is false)
netmask: "255.255.255.0"  # Netmask (if use_dhcp is false)
gateway: "192.168.1.1"  # Default gateway (if use_dhcp is false)

# DNS Configuration
dns_servers:
  - "8.8.8.8"
  - "8.8.4.4"

# Hostname Configuration
set_hostname: true  # Whether to change the hostname
hostname: "network-node"  # The desired hostname

# SSH Configuration
disable_root_login: true  # Disable root login via SSH (true/false)
ssh_port: 22  # SSH port number (default: 22)

# Firewall Configuration
firewall_enable: true  # Enable or disable firewall (true/false)
open_ports:
  - 22   # Open SSH port
  - 80   # Open HTTP port
  - 443  # Open HTTPS port

# NTP Configuration
enable_ntp: true  # Enable NTP service for time synchronization
ntp_servers:
  - "time.google.com"
  - "time.nist.gov"
```

3. user-management

Purpose: Creates users on the target servers.

Variables (`vars/main.yml`):

```
users:
  - name: devops
    password: "password123"
  - name: admin
    password: "adminpassword"
```

4. package-management

Purpose: Installs necessary software packages.

Variables (`vars/main.yml`):

```
---
```

```
# vars file for package-management
install_packages:
  - git
  - vim
  - curl
  - wget

remove_packages:
  - nmap
```

5. system-tuning

Purpose: Optimizes system performance.

Variables (`vars/main.yml`):

```
---
# vars file for system-tuning
sysctl_settings:
  - name: net.ipv4.ip_forward
    value: 1
  - name: vm.swappiness
    value: 40
```

6. monitoring

Purpose: Sets up monitoring tools.

Tasks (`tasks/main.yml`):

```
- name: Install monitoring tools
  ansible.builtin.package:
    name:
      - netdata
      - htop
      - iftop
    state: present
```

7. server-partitioning

Purpose: Manages disk partitions.

Variables (`vars/main.yml`):

```
# Note: mentioned disk should be in just attached (unmounted) status
# This Ansible role will automatically create mount points, so do not need to pre-create
# Caution: If the script fails, check the etc/fstab output before restart the system
---
disk_device: "/dev/vdb"  # Specify which disk to partition

disk_volumes:
  - { name: "installs", size: 1G, mount_point: "/installs" }
  - { name: "log", size: 1G, mount_point: "/log" }
  - { name: "backup", size: 2G, mount_point: "/backup" }
```

8. chrony_config

Purpose: Configures time synchronization.

Variables (`vars/main.yml`):

```
---
# vars file for chrony_config
```

```
# /playbooks/roles/chrony_config/vars/main.yml

chrony_servers:
  - "0.centos.pool.ntp.org"
  - "1.centos.pool.ntp.org"
  - "2.centos.pool.ntp.org"
  - "3.centos.pool.ntp.org"

timezone: "Asia/Colombo"
```

9. selinux-configuration

Purpose: Configures SELinux.

Variables (`vars/main.yml`):

```
---
# vars file for selinux-configuration
# vars/main.yml
selinux_status: enforcing  # Options: enforcing, permissive, disabled
```

10. yum_update_and_sync

Purpose: Updates system packages.

Tasks (`tasks/main.yml`):

```
# /playbooks/roles/yum_update_and_sync/tasks/main.yml
---
- name: Update all packages without prompting
  become: true
  ansible.builtin.command: yum -y update
  register: update_result

- name: Sync all yum repositories
  become: true
  ansible.builtin.command: yum makecache --refresh
  register: repo_sync_result

- name: Display yum update result
  debug:
    msg: "Yum update result: {{ update_result.stdout }}"

- name: Display yum repo sync result
  debug:
    msg: "Yum repository sync result: {{ repo_sync_result.stdout }}"
```

Running the Playbook

To execute the playbook, use:

1. Edit dev.ini file to add on which servers this playbook should run
   `Vi playbooks/dev.ini`
2. Edit site.yml to configure which roles should run on targeted servers
   `Vi playbooks/site.yml`
3. Run the playbook
   `ansible-playbook -i playbooks/dev.ini playbooks/site.yml --ask-become-pass`
4. Run the playbook – with extra verbosity
   `ansible-playbook -i playbooks/dev.ini playbooks/site.yml –ask-become-pass -vvv`