Automating System Configuration and Management Using Ansible : A Comprehensive Hands-On Project

Introduction

In today's rapidly evolving IT landscape, automation plays a crucial role in streamlining system administration, infrastructure management, and deployment processes. Ansible, a powerful open-source automation tool, is widely used to configure systems, manage applications, and orchestrate complex workflows. This assignment focuses on installing and configuring Ansible, setting up a virtual machine, managing users, installing software packages, and securing network configurations using Ansible playbooks.

The purpose of this assignment is to develop hands-on experience with Ansible, which is an essential skill for DevOps and Site Reliability Engineering (SRE) roles. By working through these tasks, I will gain a deeper understanding of infrastructure automation, configuration management, and remote system administration. Additionally, this experience will reinforce best practices in security, system administration, and efficient IT operations, which are vital for organizations that rely on scalable and automated infrastructure solutions.

This document outlines the step-by-step approach followed in the assignment, covering:

1. **Installation of Ansible** – Setting up Ansible on a local machine and ensuring it is ready for automation tasks.
2. **Virtual Machine and User Configuration** – Creating a virtual machine, adding a new sudo user, and setting up SSH key-based authentication for secure access.
3. **Inventory Management and Host Configuration** – Adding the virtual machine to the Ansible inventory and testing connectivity.
4. **User Management via Ansible Playbooks** – Creating and managing user accounts on the remote server using a structured playbook.
5. **Software Package Installation** – Automating the installation of essential system utilities and networking tools.
6. **Firewall Configuration** – Implementing network security measures using Ansible to control access to the virtual machine.

By completing these steps, I have acquired critical skills in infrastructure automation, which are directly applicable to real-world DevOps and IT operations roles. The assignment has provided an opportunity to demonstrate proficiency in writing Ansible playbooks, managing Linux systems, and automating repetitive tasks—key competencies that organizations seek in DevOps engineers.

1. **Install Ansible in your PC.**

Install EPEL repository

sudo dnf install epel-release –y

Enable the CodeReady Linux Builder repository (needed for CentOS 9 Stream)

sudo dnf config-manager --set-enabled crb

Update the repository

sudo dnf update –y

2. **Create a virtual machine and create a sudo user.**

Create a Sudo User

sudo adduser ansible_user
sudo passwd ansible_user

Grant Sudo Privileges

sudo usermod -aG wheel ansible_user
sudo visudo

ansible_user ALL=(ALL) NOPASSWD: ALL

3. **Create ansible inventory and add created VM to group 'webserver'**

- **Generate Public SSH key**

ssh-keygen

- Copy the generated key to host(remote) machine

scp ~/.ssh/id_rsa.pub root@192.168.2[redacted]/root/.ssh/authorized_keys

- SSH into the remote machine and set correct permissions

  ssh root@192.168.[redacted]
  chmod 600 ~/.ssh/authorized_keys
  exit

- Create the Ansible Inventory File

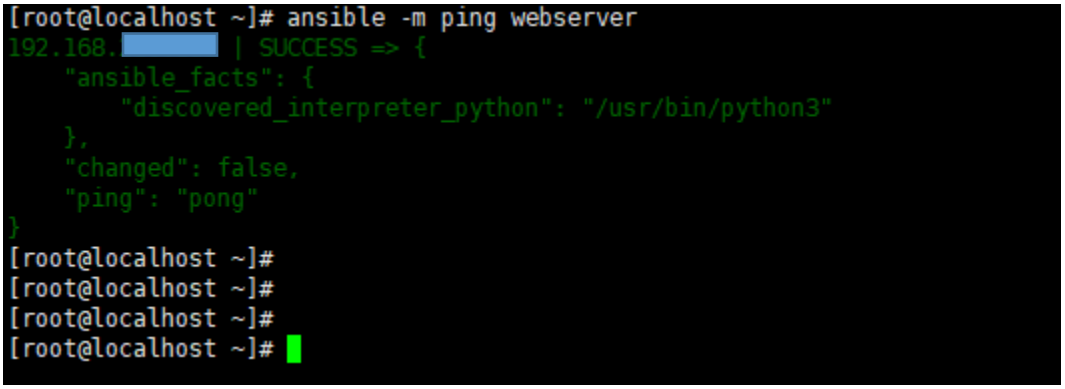
  sudo nano /etc/ansible/hosts

- Add the Target VM to the Inventory File

  [webserver]
  192.168.[redacted]

- Test Connectivity to the Target VM

  ansible -m ping webserver



4. Create ansible playbook for following

- Create and Prepare users.txt file

nano users.txt

user1,imp_user,true,4096,8192
user2,imp_user,true,2048,8192

- Copy `users.txt` to the Remote VM
  scp users.txt root@192.168.[redacted]:/tmp/users.txt

- Verification

  ssh root@192.168.[redacted] "ls -l /tmp/users.txt"

- Install the Required Ansible Collection

  ansible-galaxy collection install ansible.posix

- Create the Ansible Playbook

  nano create_users.yml

- Run the Ansible Playbook

  ansible-playbook create_users.yml

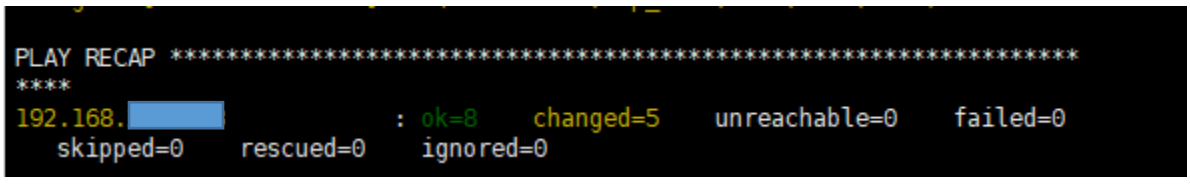
Commands That runs on Remote Machine(Host)

- Final Confirmation

  ssh root@192.168.[redacted] "cat /etc/passwd | grep user"

- Check user limits

  ssh root@192.168.[redacted] "cat /etc/security/limits.conf | grep user"



5. Install following software packages in the VM using ansible a. tcpdump, curl, wget, screen, dos2unix, nmap, ifconfig, ntp

- Create the `install_software.yml` Playbook

  nano install_software.yml

- Run the Playbook

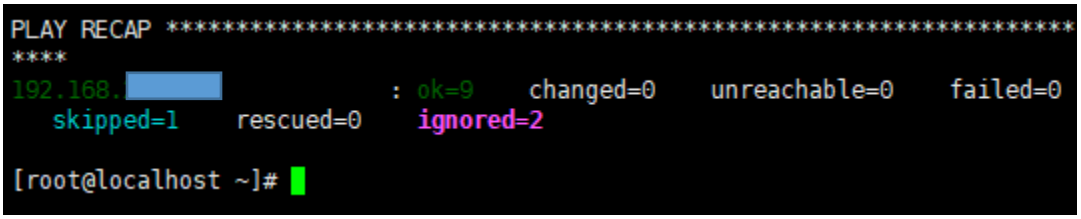
  ansible-playbook install_software.yml

- Verify Installed Packages on the Remote Server

  ssh root@192.168.[redacted] "rpm -q tcpdump curl wget dos2unix nmap net-tools tmux chrony"

- Start and Enable Chrony (NTP Alternative)

  ssh root@192.168[redacted] "systemctl enable --now chronyd"



6. Setup firewall in VM opening ssh access to any IP in Mobile LAN, port 80 and 443 from any address using ansible.

- Create **`configure_firewall.yml`**

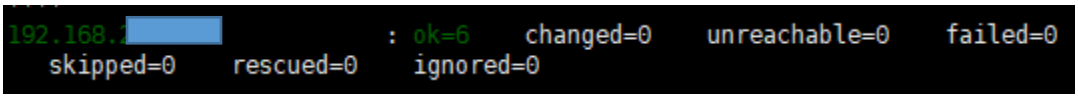
  nano configure_firewall.yml

- Run the playbook

  ansible-playbook configure_firewall.yml

- Verify the Firewall rools(host VM)

  ssh root@192.168.[redacted] "firewall-cmd --list-all"



Conclusion

Through this assignment, I have gained valuable hands-on experience with Ansible and developed essential automation skills. The key takeaways from this project include:

- **Practical Understanding of Ansible**: I have learned how to install, configure, and use Ansible effectively to manage remote servers.
- **Infrastructure as Code (IaC)**: The ability to automate system administration tasks using Ansible playbooks enhances efficiency and consistency.
- **User and Access Management**: I have implemented best practices in managing system users and secure SSH authentication.
- **Software Deployment Automation**: Installing and configuring software packages efficiently using Ansible playbooks.
- **Network Security and Firewall Configuration**: Implementing security measures to restrict and control network access to the system.
- **Troubleshooting and Debugging**: The assignment has improved my ability to troubleshoot and resolve connectivity and permission issues in an automated environment.
- **Job Readiness for DevOps Roles**: This project has strengthened my expertise in Ansible and automation, making me better prepared for DevOps and SRE job opportunities.

Overall, this assignment has reinforced my ability to work with Ansible in real-world scenarios, providing a strong foundation for infrastructure automation. Moving forward, I plan to enhance my knowledge by exploring advanced Ansible modules, integrating it with cloud platforms, and contributing to open-source automation projects. This experience has been instrumental in preparing me for my career in DevOps and automation engineering.