**Ansible Automation Scripts - Chapter 13**

**Introduction**

Ansible is a powerful open-source automation tool that simplifies system administration tasks. This document provides structured Ansible scripts and code examples extracted from Chapter 13, covering:

* Installing and Configuring Ansible
* Writing Ansible Playbooks
* Managing Infrastructure with Ansible
* Automating Rocky Linux Deployments in AWS
* Ansible Tower for Large-Scale Automation

**1. Installing and Configuring Ansible**

**1.1 Installing Ansible**

sudo dnf update -y

sudo dnf install ansible -y

ansible --version

**1.2 Configuring Ansible Inventory**

Edit the inventory file (/etc/ansible/hosts) and add managed nodes:

[webservers]

web1 ansible\_host=192.168.1.10 ansible\_user=admin

web2 ansible\_host=192.168.1.11 ansible\_user=admin

dbservers]

db1 ansible\_host=192.168.1.12 ansible\_user=admin

[all\_servers:children]

webservers

dbservers

**1.3 Setting Up SSH Key Authentication**

Generate SSH keys and copy them to remote hosts:

ssh-keygen -t rsa -b 4096

ssh-copy-id admin@192.168.1.10

ssh-copy-id admin@192.168.1.11

ssh-copy-id admin@192.168.1.12

Verify SSH connection:

ssh admin@192.168.1.10

ssh admin@192.168.1.11

ssh admin@192.168.1.12

Test connectivity using Ansible:

ansible all -m ping

**2. Writing Ansible Playbooks**

**2.1 Creating a Simple Playbook**

Create a file setup-webserver.yml:

---

- name: Setup Web Server

hosts: webservers

become: yes

tasks:

- name: Install Nginx

yum:

name: nginx

state: present

- name: Start Nginx

service:

name: nginx

state: started

enabled: yes

Run the playbook:

ansible-playbook setup-webserver.yml --ask-become-pass

**3. Managing Infrastructure with Ansible**

**3.1 Provisioning AWS EC2 Instance**

---

- name: Provision EC2 Instance

hosts: localhost

tasks:

- name: Launch EC2 Instance

ec2:

key\_name: my-key

instance\_type: t2.micro

image: ami-0abcdef1234567890

wait: yes

region: us-west-2

group: my-security-group

count: 1

register: ec2

**3.2 Automating Rocky Linux Deployments in AWS**

Create deploy-rocky-app.yml:

---

- name: Deploy Rocky Linux Application

hosts: rocky\_servers

become: yes

tasks:

- name: Install Required Packages

yum:

name:

- httpd

- git

state: present

- name: Start Apache

service:

name: httpd

state: started

enabled: yes

Run the playbook:

ansible-playbook deploy-rocky-app.yml

**4. Ansible Tower for Large-Scale Automation**

**4.1 Installing Ansible Tower**

sudo ./setup.sh

Access Tower via Web UI:

http://<tower\_server\_ip>/

**4.2 Creating a Job Template in Ansible Tower**

1. Go to **Job Templates**.
2. Define job parameters:
   * **Name**: Deploy Rocky Linux
   * **Inventory**: rocky\_servers
   * **Playbook**: deploy-rocky-app.yml
   * **Credentials**: SSH Key
3. Save and launch the job.

**Conclusion**

These Ansible scripts help streamline automation for Rocky Linux deployments in AWS and infrastructure management. By implementing these structured playbooks, administrators can reduce manual intervention and ensure consistent configurations.