**ANSIBLE SETUP:**

* Install ansible-core

>>yum install ansible-core

* Add ansible user

>>useradd ansible

* Set password for the user

>>passwd ansible --ansible

* Add in sudoers file

>>ansible ALL=(ALL) NOPASSWD=ALL

* Add in /etc/ssh/sshd\_config

>>/etc/ssh/sshd\_config publickey authentication yes | root login on

- >> systemctl restart sshd

-Key generation:

>> ssh-keygen

- >> ssh-copyid root@192.168.3.203 --automatically will have authorized keys

add in ansible host file

ansible all --list-hosts

- Collections (Additional)

- install galaxy collection module offine

>>ansible-galaxy collection install community.general -p ./collections

ANSIBLE PLAYBOOK: For JBoss Linux

---  
 - name: Deploy and Start JBoss  
 hosts: dev  
 become: yes

tasks:  
 - name: Ensure JBoss user exists  
 ansible.builtin.user:  
 name: jboss  
 system: yes  
 create\_home: yes  
 home: /opt/jboss

- name: Set ownership for JBoss directory  
 ansible.builtin.file:  
 path: /opt/jboss-eap-7.4  
 state: directory  
 recurse: yes  
 owner: jboss  
 group: jboss

- name: Check if Java is already installed  
 stat:  
 path: /opt/jdk-17.0.12  
 register: Java\_installed

- name: Copy Java tar to remote host  
 copy:  
 src: /etc/ansible/jdk-17\_linux-x64\_bin.tar.gz  
 dest: /opt/jdk-17\_linux-x64\_bin.tar.gz

- name: Install Java  
 unarchive:  
 src: /opt/jdk-17\_linux-x64\_bin.tar.gz  
 dest: /opt  
 remote\_src: yes

- name: Check if JBoss is already installed  
 stat:  
 path: /opt/jboss-eap-7.4  
 register: jboss\_installed

- name: Copy JBoss ZIP to remote host  
 copy:  
 src: /etc/ansible/jboss-eap-7.4.0.zip  
 dest: /opt/jboss-eap-7.4.0.zip  
 when: not jboss\_installed.stat.exists

- name: Install JBoss  
 unarchive:  
 src: /opt/jboss-eap-7.4.0.zip  
 dest: /opt  
 remote\_src: yes  
 mode: '0644'  
 when: not jboss\_installed.stat.exists

- name: Create symbolic link to JBoss  
 file:  
 src: /opt/jboss-eap-7.4  
 dest: /opt/jboss  
 state: link  
 when: not jboss\_installed.stat.exists

- name: Copy JBoss service file  
 ansible.builtin.template:  
 src: templates/jboss.service.j2  
 dest: /etc/systemd/system/jboss.service  
 register: jboss\_service

- name: Reload systemd  
 ansible.builtin.systemd:  
 daemon\_reload: yes

- name: Ensure JBoss service is enabled and started  
 ansible.builtin.systemd:  
 name: jboss  
 enabled: yes  
 state: started

**Ansible playbook (Using jdk 1.8.0\_202 )**

**Prerequisite:** ansible-galaxy collection install ansible.posix

---

- hosts: dev

user: ansible

become: yes

gather\_facts: yes

tasks:

- name: Ensure JBoss user exists

ansible.builtin.user:

name: jboss

system: yes

create\_home: yes

home: /opt/jboss

- name: Set ownership for JBoss directory

ansible.builtin.file:

path: /opt/jboss-eap-7.4

state: directory

recurse: yes

owner: jboss

group: jboss

- name: Check if Java is already installed

stat:

path: /opt/jdk1.8.0\_202

register: Java\_installed

- name: Copy Java tar to remote host

copy:

src: /etc/ansible/clos/jdk-8u202-linux-x64.tar.gz

dest: /opt/jdk-8u202-linux-x64.tar.gz

- name: Install Java

unarchive:

src: /opt/jdk-8u202-linux-x64.tar.gz

dest: /opt

remote\_src: yes

mode: '0644'

- name: Check if JBoss is already installed

stat:

path: /opt/jboss-eap-7.4

register: jboss\_installed

- name: Copy JBoss ZIP to remote host

copy:

src: /etc/ansible/clos/jboss-eap-7.4.0.zip

dest: /opt/jboss-eap-7.4.0.zip

when: not jboss\_installed.stat.exists

- name: Install JBoss

unarchive:

src: /opt/jboss-eap-7.4.0.zip

dest: /opt

remote\_src: yes

mode: '0644'

when: not jboss\_installed.stat.exists

- name: Create symbolic link to JBoss

file:

src: /opt/jboss-eap-7.4

dest: /opt/jboss

state: link

when: not jboss\_installed.stat.exists

- name: Copy JBoss service file

ansible.builtin.template:

src: templates/jboss.service.j2

dest: /etc/systemd/system/jboss.service

register: jboss\_service

- name: Reload systemd

ansible.builtin.systemd:

daemon\_reload: yes

- name: Ensure JBoss service is enabled and started

ansible.builtin.systemd:

name: jboss

enabled: yes

state: started

# New tasks for firewall configuration

- name: Install firewalld

ansible.builtin.yum:

name: firewalld

state: latest

notify:

- start firewalld

- name: Ensure firewalld is started and enabled

ansible.builtin.service:

name: firewalld

state: started

enabled: yes

- name: Open port 8080 in firewalld

ansible.posix.firewalld:

zone: public

port: 8080/tcp

permanent: yes

state: enabled

- name: Reload firewalld to apply changes

ansible.builtin.service:

name: firewalld

state: reloaded

- name: Copy WAR file to JBoss deployment directory

copy:

src: /etc/ansible/sample.war

dest: /opt/jboss-eap-7.4/standalone/deployments/

mode: '0644'

- name: Stop JBoss service after deployment

ansible.builtin.systemd:

name: jboss

state: stopped

-------------------------------------------------------------------------------------------------------------

**SERVICE FILE : PATH** (etc/ansible/templates/ jboss.service.j2)

[Unit]

Description=JBoss Application Server

After=network.target

[Service]

Type=simple

User=jboss

Group=jboss

Environment="JAVA\_HOME=/opt/jdk1.8.0\_202"

ExecStart=/opt/jboss-eap-7.4/bin/standalone.sh -b 0.0.0.0 -bmanagement 0.0.0.0

ExecStop=/opt/jboss-eap-7.4/bin/jboss-cli.sh --connect --command=:shutdown

Restart=on-failure

RestartSec=10

[Install]

WantedBy=multi-user.target

**Ansible-Tomcat-deploy:**

**Tomcat.yml**

**-** hosts: dev

user: ansible

become: yes

gather\_facts: yes

tasks:

- name: Ensure Tomcat user exists

ansible.builtin.user:

name: tomcat

system: yes

create\_home: yes

home: /opt/tomcat

- name: Set ownership for Tomcat directory

ansible.builtin.file:

path: /opt/tomcat

state: directory

recurse: yes

owner: tomcat

group: tomcat

- name: Check if Java is already installed

stat:

path: /opt/java-1.8.0-openjdk-portable-1.8.0.402.b06-1.portable.jdk.el.x86\_64.tar.xz

register: Java\_installed

- name: Copy Java tar to remote host

copy:

src: /etc/ansible/posidex/java-1.8.0-openjdk-portable-1.8.0.402.b06-1.portable.jdk.el.x86\_64.tar.xz

dest: /opt

- name: Install Java

unarchive:

src: /opt/java-1.8.0-openjdk-portable-1.8.0.402.b06-1.portable.jdk.el.x86\_64.tar.xz

dest: /opt

remote\_src: yes

mode: '0755'

- name: Ensure JAVA\_HOME is set correctly

lineinfile:

path: ~/.bash\_profile

line: 'export JAVA\_HOME=/opt/java-1.8.0-openjdk-portable-1.8.0.402.b06-1.portable.jdk.el.x86\_64'

create: yes

- name: Check if Tomcat is already installed

stat:

path: /opt/apache-tomcat-9.0.91

register: tomcat\_installed

- name: Copy Tomcat ZIP to remote host

copy:

src: /etc/ansible/posidex/apache-tomcat-9.0.91.zip

dest: /opt/apache-tomcat-9.0.91.zip

when: not tomcat\_installed.stat.exists

- name: Install Tomcat

unarchive:

src: /opt/apache-tomcat-9.0.91.zip

dest: /opt

remote\_src: yes

mode: '0777'

when: not tomcat\_installed.stat.exists

- name: Set ownership and permissions for Tomcat directory

ansible.builtin.file:

path: /opt/apache-tomcat-9.0.91/\*

state: directory

recurse: yes

mode: '0777'

- name: Remove stale PID file if exists

ansible.builtin.command:

cmd: rm -rf /opt/apache-tomcat-9.0.91/temp/tomcat.pid

- name: Set permission to catalina.out

ansible.builtin.command:

cmd: chmod 777 /opt/apache-tomcat-9.0.91/logs/catalina.out

- name: Copy Tomcat service file

ansible.builtin.template:

src: templates/tomcat.service.j2

dest: /etc/systemd/system/tomcat.service

register: tomcat\_service

- name: Reload systemd

ansible.builtin.systemd:

daemon\_reload: yes

- name: Ensure Tomcat service is enabled and started

ansible.builtin.systemd:

name: tomcat

enabled: yes

state: started

# New tasks for firewall configuration

- name: Install firewalld

ansible.builtin.yum:

name: firewalld

state: latest

notify:

- start firewalld

- name: Ensure firewalld is started and enabled

ansible.builtin.service:

name: firewalld

state: started

enabled: yes

- name: Open port 8080 in firewalld

ansible.posix.firewalld:

zone: public

port: 8080/tcp

permanent: yes

state: enabled

- name: Reload firewalld to apply changes

ansible.builtin.service:

name: firewalld

state: reloaded

- name: Copy WAR file to Tomcat webapps directory

copy:

src: /etc/ansible/sample.war

dest: /opt/apache-tomcat-9.0.91/webapps/ROOT

mode: '0644'

**tomcat.service.j2**

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

# Environment variables

Environment=JAVA\_HOME=/opt/java-1.8.0-openjdk-portable-1.8.0.402.b06-1.portable.jdk.el.x86\_64

Environment=CATALINA\_PID=/opt/apache-tomcat-9.0.91/temp/tomcat.pid

Environment=CATALINA\_HOME=/opt/apache-tomcat-9.0.91

Environment=CATALINA\_BASE=/opt/apache-tomcat-9.0.91

Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'

Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

# ExecStart and ExecStop paths

ExecStart=/opt/apache-tomcat-9.0.91/bin/startup.sh

ExecStop=/opt/apache-tomcat-9.0.91/bin/shutdown.sh

# User and group

User=tomcat

Group=tomcat

# Allow the service to restart automatically in case of failure

RestartSec=10

Restart=on-failure

[Install]

WantedBy=multi-user.target

For Posidex app :  
1.apache-tomcat-9.0.91.zip

2.java-1.8.0-openjdk-portable-1.8.0.402.b06-1.portable.jdk.el.x86\_64.tar.xz

3. ansible.posix.firewalld module