**Write and Ansible playbook to install configure and deploy SonarQube on RHEL 9**

Setting up SonarQube with Ansible involves automating the installation and configuration of SonarQube using Ansible playbooks. This approach streamlines the deployment process, ensuring consistency and reproducibility across different environments.

Install SonarQube from a ZIP file or use Docker. Ansible can handle tasks like installing Java, downloading SonarQube, configuring the database (if not using embedded), and setting up SonarQube as a service.

Infrastructure as code approach to ensure the new server will be maintainable for years to come.

Diagram that shows what we’ll be building:

A blue screen with white text and symbols

AI-generated content may be incorrect.

**My infrastructure as code philosophy**

My day job as systems engineer only occasionally allows me to play with infrastructure, so my toolbox is limited to what I’ve been exposed in my 12 years of working to or have had time to learn on my own:

* **Linux:** 15+ years of experience, mostly Red Hat and CentOS flavors
* **Docker:** 4+ years of experience, not a master, but I haven’t yet run into any problems I couldn’t surmount.
* **Ansible:** 5+ years of experience, again not a master, but I’ve been able to solve most problems I’ve run across.

**Create an Ansible playbook to install configure and deploy:**

# su - hogege

# cd ansible\_automation

# vim install\_sonarqube.yaml

---

- name: Install and configure SonarQube on RHEL 9

hosts: sonarqube

become: true

vars:

vars:

sonar\_version: "25.6.0.109173"

sonar\_user: sonar

sonar\_group: sonar

sonar\_home: /opt/sonarqube

postgres\_password: "password"

db\_name: sonarqube

db\_user: sonar

db\_password: "sonar"

postgresql\_version: 16

postgresql\_service: "postgresql-16"

tasks:

- name: Install SonarQube dependencies packages

dnf:

name:

- wget

- unzip

state: present

- name: Create a directory java\_package using Ansible playbook

file: path=/root/java\_package state=directory

- name: Copy java jdk 17 rpm file to remote or target server

copy:

src: /root/java\_package/jdk-17.0.13\_linux-x64\_bin.rpm

dest: /root/java\_package/jdk-17.0.13\_linux-x64\_bin.rpm

- name: Installed java 17 rpm file

args:

chdir: /root/java\_package

shell: dnf localinstall jdk-17.0.13\_linux-x64\_bin.rpm -y

- name: Install and Configure the PostgreSQL RPM Repository

dnf:

name: https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-x86\_64/pgdg-red

hat-repo-latest.noarch.rpm

state: present

disable\_gpg\_check: true

- name: Disable the default PostgreSQL module once the repository is added

shell: yum -qy module disable postgresql

- name: Install the PostgreSQL 16 database server

dnf:

name:

- postgresql{{ postgresql\_version }}

- postgresql{{ postgresql\_version }}-server

- python3-psycopg2

state: present

- name: Initialize Postgres Database

command: "/usr/pgsql-{{ postgresql\_version }}/bin/postgresql-{{ postgresql\_version

}}-setup initdb"

args:

creates: "/var/lib/pgsql/16/data/PG\_VERSION"

- name: Start and Enable the postgresql-16 service

systemd:

name: postgresql-{{ postgresql\_version }}

state: started

enabled: true

- name: Set PostgreSQL password and create SonarQube DB

become: true

become\_user: postgres

shell: |

psql -c "ALTER USER postgres WITH PASSWORD '{{ postgres\_password }}';"

psql -c "CREATE USER {{ db\_user }} WITH PASSWORD '{{ db\_password }}';"

psql -c "ALTER USER {{ db\_user }} WITH ENCRYPTED password '{{ db\_password }}';"

psql -c "CREATE DATABASE {{ db\_name }} OWNER {{ db\_user }};"

psql -c "GRANT ALL PRIVILEGES ON DATABASE {{ db\_name }} TO {{ db\_user }};"

psql -c "ALTER ROLE {{ db\_user }} SET client\_encoding TO 'utf8';"

psql -c "ALTER ROLE {{ db\_user }} SET default\_transaction\_isolation TO 'read comm

itted';"

psql -c "ALTER ROLE {{ db\_user }} SET timezone TO 'UTC';"

args:

executable: /bin/bash

- name: Update pg\_hba.conf for password auth

lineinfile:

path: /var/lib/pgsql/16/data/pg\_hba.conf

regexp: '^host\s+{{ db\_name }}\s+{{ db\_user }}\s+127\.0\.0\.1/32\s+'

line: "host {{ db\_name }} {{ db\_user }} 127.0.0.1/32 md5"

state: present

notify: Restart PostgreSQL

- name: Download SonarQube

get\_url:

url: "https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-{{ sonar\_

version }}.zip"

dest: /tmp/sonarqube.zip

- name: Unpack SonarQube

unarchive:

src: /tmp/sonarqube.zip

dest: /opt/

remote\_src: yes

creates: "/opt/sonarqube-{{ sonar\_version }}"

- name: Symlink to /opt/sonarqube

file:

src: "/opt/sonarqube-{{ sonar\_version }}"

dest: "{{ sonar\_home }}"

state: link

force: true

- name: Create SonarQube user

user:

name: "{{ sonar\_user }}"

shell: /sbin/nologin

system: true

home: "{{ sonar\_home }}"

- name: Set ownership of SonarQube

file:

path: "{{ sonar\_home }}"

state: directory

recurse: yes

owner: "{{ sonar\_user }}"

group: "{{ sonar\_group }}"

- name: Configure SonarQube database

lineinfile:

path: "{{ sonar\_home }}/conf/sonar.properties"

regexp: '^#?sonar.jdbc.url='

line: "sonar.jdbc.url=jdbc:postgresql://localhost/{{ db\_name }}"

notify: Restart SonarQube

- name: Configure SonarQube username and password

blockinfile:

path: "{{ sonar\_home }}/conf/sonar.properties"

block: |

sonar.jdbc.username={{ db\_user }}

sonar.jdbc.password={{ db\_password }}

- name: Create SonarQube systemd service

copy:

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

content: |

[Unit]

Description=SonarQube service

After=syslog.target network.target

[Service]

Type=forking

ExecStart={{ sonar\_home }}/bin/linux-x86-64/sonar.sh start

ExecStop={{ sonar\_home }}/bin/linux-x86-64/sonar.sh stop

User={{ sonar\_user }}

Group={{ sonar\_group }}

Restart=always

LimitNOFILE=65536

LimitNPROC=4096

[Install]

WantedBy=multi-user.target

- name: Reload systemd

systemd:

daemon\_reload: yes

- name: Enable and start SonarQube

systemd:

name: sonarqube

enabled: yes

state: started

- name: Set sysctl values for Elasticsearch

sysctl:

name: "{{ item.name }}"

value: "{{ item.value }}"

state: present

sysctl\_set: yes

reload: yes

loop:

- { name: "vm.max\_map\_count", value: "262144" }

- { name: "fs.file-max", value: "65536" }

- name: Set limits for SonarQube user

blockinfile:

path: /etc/security/limits.conf

block: |

sonarqube - nofile 65536

sonarqube - nproc 4096

marker: "# {mark} ANSIBLE MANAGED SONARQUBE LIMITS"

handlers:

- name: Start and Enable the postgresql-16 service

systemd:

name: postgresql-{{ postgresql\_version }}

state: started

- name: Restart SonarQube

systemd:

name: sonarqube

state: restarted

# sudo ansible-playbook install\_sonarqube.yaml --syntax-check

# sudo ansible-playbook install\_sonarqube.yaml

**Post-Install Steps**

After running the playbook:

Visit: <http://10.0.0.45:9000>

default credentials: **admin / admin**

Change the password on first login.