

SONARQUBE

Sandoval Lucas , Leutwyler Nicolás

1er cuatrimestre 2015

1. Introducción

1.1. Descripción del proyecto

El proyecto abarca la idea de ofrecer un servicio de análisis de código que permita analizar diferentes lenguajes, con distintos analizadores y con distintos criterios. En particular usaremos SonarQube Runner como analizador, y dos configuraciones de reglas distintas.

1.2. Relevamiento de características de la máquina utilizada

Procesador: Intel Core i3 CPU M 380 @ 2.53GHz x 2

RAM: 3.7 Gb

HD: 613.1 Gb

1.3. Relevamiento del Sistema operativo

Sistema operativo: Ubuntu 14.04.2 LTS

Desktop: Cinnamon version 2.4.8

Kernel: 3.13.0-53-generic

1.4. Software utilizado y/o necesario

- MySQL 5.5.43, for debian-linux-gnu (i686).
- Apache2 webserver 2.4.7 Ubuntu
- java-7-openjdk-i386
- Mozilla firefox

2. SonarQube - Breve descripción

Sonarqube es una plataforma de código abierto (Software Libre) para la verificación y mantenimiento de la calidad de código. Esta herramienta nos provee la cobertura de los 7 pilares dentro de la calidad del código:

- Arquitectura y Diseño
- Comentarios
- Reglas de Código
- Errores potenciales
- Complejidad
- Tests de Unidad
- Duplicaciones

3. Empezando

3.1. Estructura del sistema

- Se necesita una base de datos para:
 - La configuración de la instancia de SonarQube(seguridad,plugins,etc).
 - La calidad de snapshots para los proyectos, vistas, etc.
- Un único servidor web para los usuarios.
- Uno o mas *Analizadores* para utilizar.

3.2. Consideraciones

Algunas consideraciones a tener en cuenta:

- La plataforma no puede tener mas de un servidor web o mas de una base de datos.
- Para rendimiento óptimo, cada componente debería estar instalado en un equipo dedicado. Sin embargo el servidor web y la base de datos, pueden estar instalados en la misma máquina.
- Analizadores escalarán al agregar equipos.
- La base de datos y los analizadores deben estar localizadas en la misma red.
- La base de datos y los analizadores no deberian estar instalados en la misma máquina.
- Todas los equipos deben estar sincronizados respecto al tiempo.

4. Instalación

4.1. Base de datos

Primero vamos a instalar la base de datos, con el motor innoDB haciendo los comandos:

- `sudo apt-get install mysql-server`
- `sudo apt-get install innodb`

Una vez que tenemos esto instalado procedemos a crear la base de datos que sonar utilizará y que luego configuraremos. Para empezar el proceso primero hacemos `mysql -user=root -password=root` (cuando se instala mysql desde `apt-get install` hay que configurar un user y un password, que en nuestro caso es `root - root`) desde la consola y seguimos los pasos:

- `CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8_general_ci`
- `CREATE USER 'sonarqube' IDENTIFIED BY 'sonarqube'`
- `GRANT ALL ON sonar.* TO 'sonar'@'%' IDENTIFIED BY 'sonarqube'`

En este punto ya tenemos la base de datos lista para continuar con la instalación del resto de las cosas.

4.2. Servidor web SonarQube

Descargamos la versión 5.1.1 de la página `sonarqube.org` que viene en formato zip, luego lo descomprimos en el directorio `/opt`. Es decir que desde ahora nuestro directorio de instalación es `/opt/sonarqube-5.1.1` que desde ahora llamaremos `<sonar>`

4.2.1. Conectando con la base de datos

Hay que entrar en `<sonar>/conf/sonar.properties` y configurar las opciones de bases de datos. Hay plantillas para los 3 tipos de configuraciones y bases de datos, en nuestro caso tenemos que descomentar la parte que dice mysql. Las líneas específicas en nuestro caso son:

- `sonar.jdbc.username=sonarqube`
- `sonar.jdbc.password=sonarqube`
- `sonar.jdbc.url=jdbc:mysql://localhost:3306/sonarqube?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance`

4.2.2. Iniciando el servidor

El puerto por defecto de sonar es el 9000 y el sitio de inicio es el «/».

En nuestro caso dejamos las configuraciones por defecto, pero para cambiar estas configuraciones hay que entrar a `<sonar>/conf/sonar.properties` y editar las líneas con los datos que queramos.

- `sonar.web.host=localhost`
- `sonar.web.context=/`
- `sonar.web.port=9000`

Ahora ya estamos en condiciones de iniciar el servidor. Ejecutamos el script ubicado en `<sonar>/bin/linux-x86-32/sonar.sh start`.

4.3. Instalando un Analyzer

Dentro de los analizadores que podemos instalar en nuestro SonarQube, se encuentran:

- SonarQube Runner: recomendado para proyectos que no usen Maven.
- Maven: recomendado para proyectos creados con Maven.
- SonarQube Ant Task: para integrar proyectos hechos con Ant.
- Gradle: integrar proyectos hechos con Gradle.

En nuestro caso instalaremos el SonarQube Runner. Sonar Runner

En este caso hicimos lo mismo que con el Sonar Web Server. Descomprimos el zip descargado en `/opt` y ahora nuestro lugar de instalación es `/opt/sonarRunner` y lo llamaremos `<sonarRunner>`

Hay que crear una variable de entorno «SONAR_HOME»
Añadimos en `/etc/bash.bashrc` (ejecutando `nano /etc/bash.bashrc`) las líneas:

- `SONAR_RUNNER_HOME=<sonarRunner>`
- `export SONAR_RUNNER_HOME`
- `PATH=$PATH:$SONAR_RUNNER_HOME/bin`

5. Caso de uso

En esta sección explicaremos cómo analizar código java desde eclipse utilizando el analizador de sonarqube recién instalado

```

# must be unique in a given SonarQube instance
sonar.projectKey=my:project
# this is the name displayed in the SonarQube UI
sonar.projectName=My project
sonar.projectVersion=1.0

# Path is relative to the sonar-project.properties file. Replace "\" by "/" on Windows.
# Since SonarQube 4.2, this property is optional if sonar.modules is set.
# If not set, SonarQube starts looking for source code from the directory containing
# the sonar-project.properties file.
sonar.sources=.

# Encoding of the source code. Default is default system encoding
#sonar.sourceEncoding=UTF-8

```

Figura 1: sonar-project.properties

- Posicionarse en el directorio del proyecto a analizar
- Ejecutar `touch sonar-⟨nombreProyecto⟩.properties`
- Editar este archivo para que tenga un contenido como la Figura 1
- Ejecutar el comando `sonnar-runner`

5.1. Viendo resultados

Para ver las métricas y medidas del análisis que sonar hizo sobre nuestro código simplemente entramos en `localhost:9000` y navegamos al proyecto en cuestión.

Cualquiera en la misma red puede realizar un análisis sobre su código entrando a la ip del servidor que verificamos ejecutando el comando `ifconfig`.

6. Problemas surgidos

Uno de los problemas puede ser el puerto. En nuestro caso teníamos instalado un `jboss` que corria en el puerto `9000`, y cuando se hacía `sonar start`, no iniciaba.

Este problema puede ser detectado entrando a `localhost:9000` (en caso de ser esta la dirección en la que está corriendo el sonar) para ver si está corriendo algún otro servidor en ese puerto.

La solución fué cambiar el puerto de `jboss`, pero también se puede solucionar cambiando el puerto de sonar, entrando a `⟨sonar⟩/conf/sonar.properties`.

7. Referencias

- SonarQube WebPage
- Pasos para instalacón