

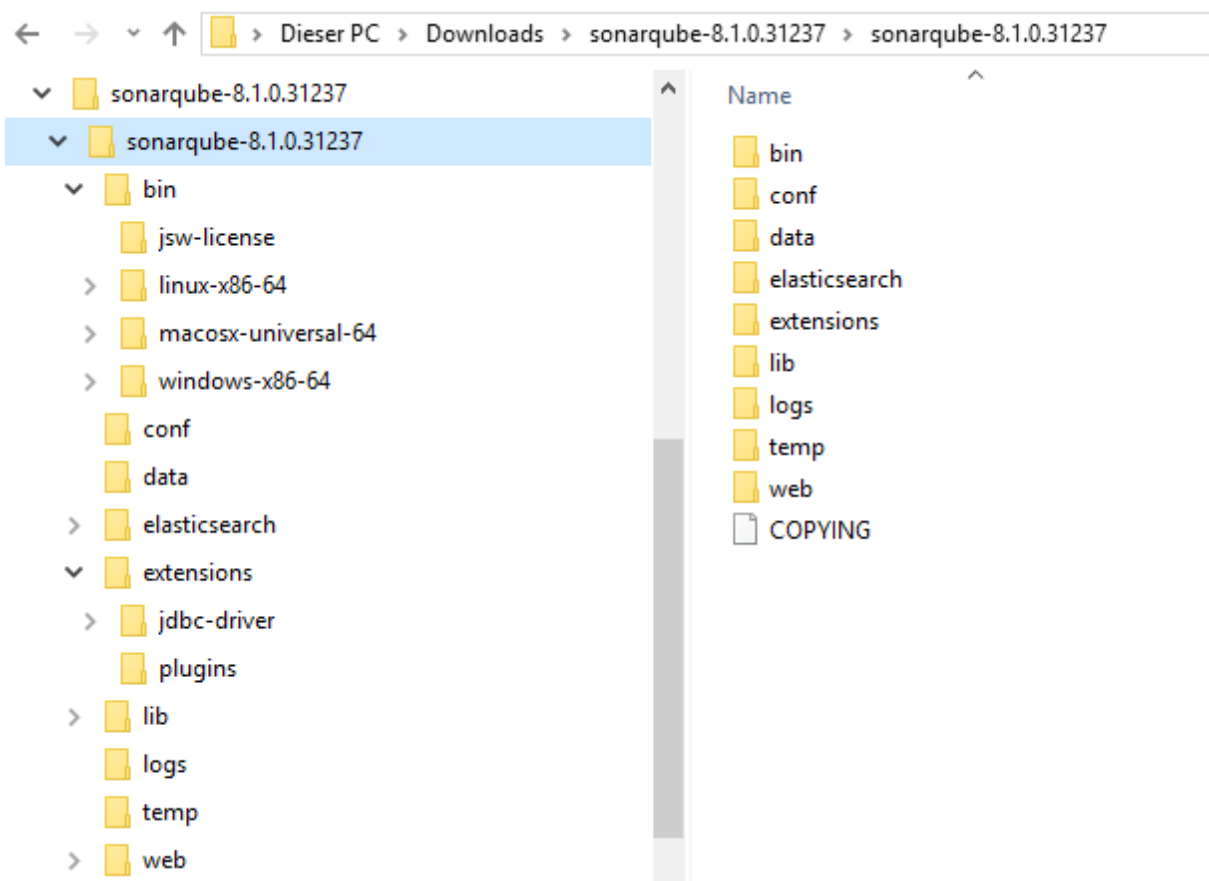
How To SonarQube

Installation von SonarQube:

Im Folgenden geht es um den Einsatz von SonarQube in C#, Java und C/C++ Projekten für Windows 10 und Ubuntu (18.04.3 LTS):

Als erstes, Es soll die SonarQube [Community Edition](#) (Diese Lizenz bezieht sich auf Version 3 der GNU Lesser General Public License) von der [offiziellen Website](#) heruntergeladen werden.

Nach dem Auspacken der gezippten aktuellen Version findet sich im Installationsverzeichnis die folgende Verzeichnisstruktur:



- In Ubuntu den Befehl eingeben, um die Zip zu downloaden:
`wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.1.0.31237.zip`

Meine Empfehlung immer neuesten Version der Anwendung suchen.

- Danach das Archiv mit dem folgenden Befehl Entpacken:
`sudo unzip sonarqube-8.1.0.31237.zip -d /opt`
- Dann das Verzeichnis Umbenennen:
`sudo mv /opt/sonarqube-8.1.0.31237.zip /opt/sonarqube`
- Die Administratorbenutzer Berechtigungen für das Verzeichnis Zuordnen:
`sudo chown -R administrator:administrator /opt/sonarqube/`
- Als Nächstes sollte SonarQube so konfiguriert werden, dass es als Sonar-Benutzer ausgeführt wird. Sie können dies mit dem folgenden Befehl tun:
`sudo adduser sonar (neu Benutzer)`
`sudo nano /opt/sonarqube/bin/linux-x86-64/sonar.sh`

Die folgenden Änderungen machen und danach speichern:

```
RUN_AS_USER=sonar
```

- Um SonarQube mit einer Datenbank zu konfigurieren, die folgenden Änderungen in SonarQube-Standardkonfigurationsdatei sind nötig, wie in diesem Beispiel:

```
GNU nano 2.9.3 /opt/sonarqube/conf/sonar.properties Verändert
#-----$
# DATABASE
#
# IMPORTANT:
# - The embedded H2 database is used by default. It is recommended for tests bu$
#   production use. Supported databases are MySQL, Oracle, PostgreSQL and Micro$
# - Changes to database connection URL (sonar.jdbc.url) can affect SonarSource $
# User credentials.
# Permissions to create tables, indices and triggers must be granted to JDBC us$
# The schema must be created first.

sonar.jdbc.username=sonar
sonar.jdbc.password=password
sonar.jdbc.url=jdbc:postgresql://localhost/sonar
sonar.web.host=127.0.0.1
sonar.search.javaOpts=-Xms512m -Xmx512m
```

- Als Nächstes sollte eine systemd-Dienstdatei erstellt werden, um den SonarQube-Dienst zu verwalten:

```
sudo nano /etc/systemd/system/sonar.service
```

```
GNU nano 2.9.3 /etc/systemd/system/sonar.service

[Unit]
Description=SonarQube service
After=syslog.target network.target

[Service]
Type=forking

ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start
ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop

User=sonar
Group=sonar
Restart=always

[Install]
WantedBy=multi-user.target
```

Nachdem Speicherung und um SonarQube Dienst zu starten, die zwei Befehle eingeben:

```
sudo systemctl start sonar
```

```
sudo systemctl enable sonar
```

Standardmäßig überwacht SonarQube Port 9000. Daher muss ein Server wie Apache als Reverse-Proxy installiert und konfiguriert werden, um über Port 80 auf den SonarQube zugreifen zu können.

Und im Folgenden geht's um die Installation des Servers mit den Befehlen:

- Apache Installation:

```
sudo apt-get install apache2
```
- Aktivierung des Moduls „mod_proxy“

```
sudo a2enmod proxy
```

```
sudo a2enmod proxy_http
```
- Apache Virtual Host-Datei für SonarQube mit dem folgenden Befehl:

```
sudo nano /etc/apache2/sites-available/sonar.conf
```

die folgenden Zeilen schreiben, um den Host zu konfigurieren.

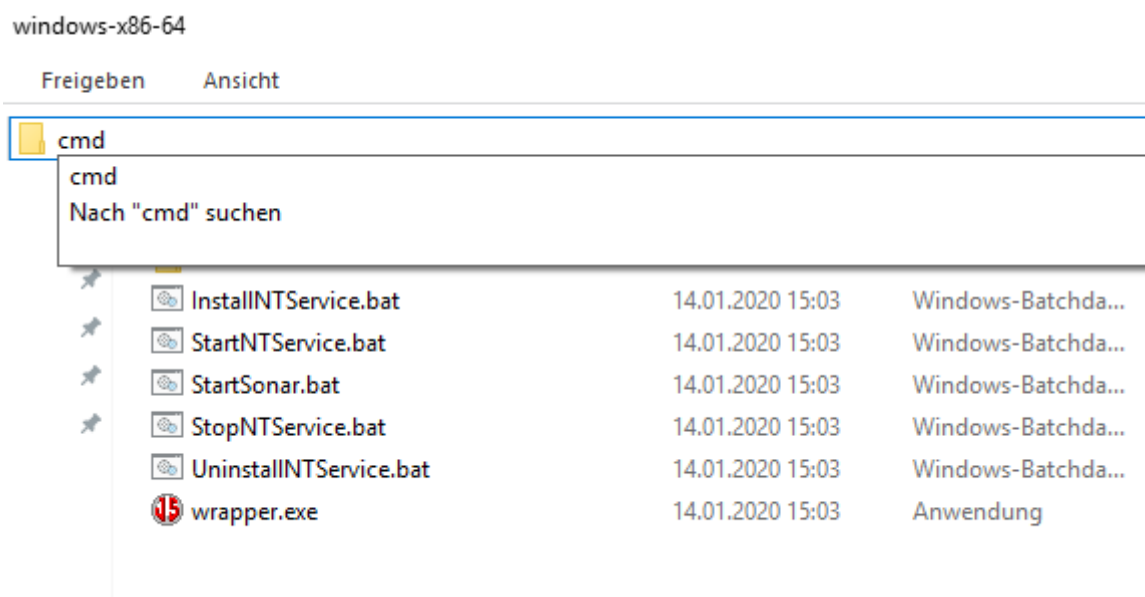
```
GNU nano 2.9.3 /etc/apache2/sites-available/sonar.conf

<VirtualHost *:80>
    ServerName sonar.mo
    ServerAdmin admin@sonar.mo
    ProxyPreserveHost On
    ProxyPass / http://127.0.0.1:9000/
    ProxyPassReverse / http://127.0.0.1:9000/
    TransferLog /var/log/apache2/sonarm_access.log
    ErrorLog /var/log/apache2/sonar_error.log
</VirtualHost>
```

Abschließend, nachdem Speicherung sollten die zwei Dienste Apache und SonarQube mit den folgenden Befehlen neugestartet:

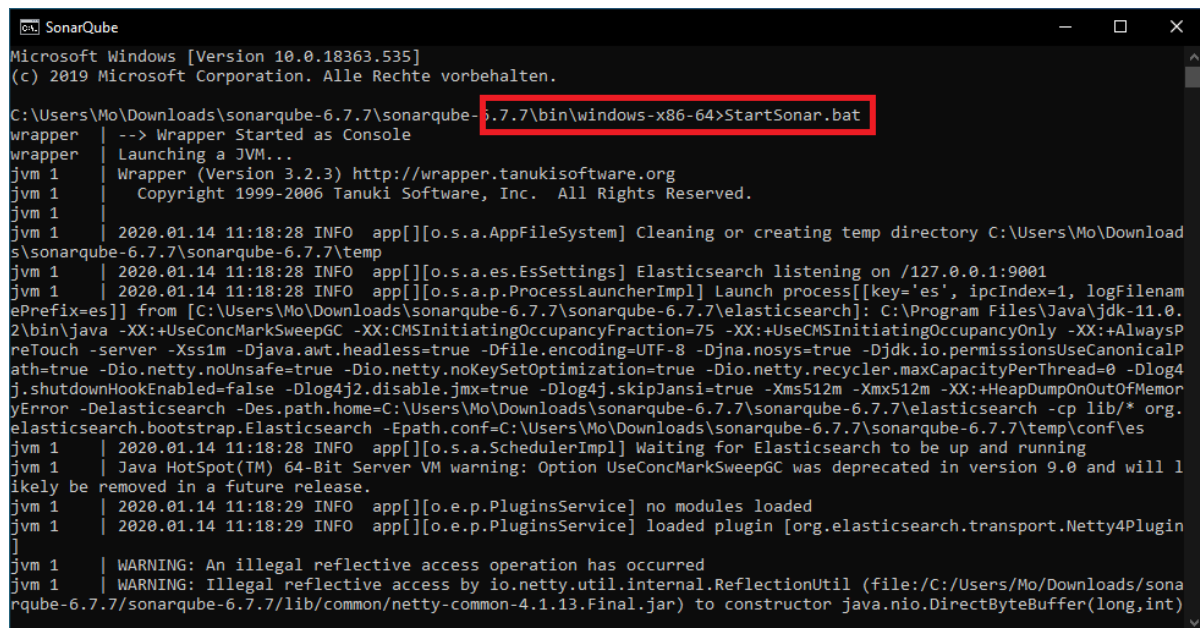
```
sudo systemctl restart apache2
sudo systemctl restart sonar
```

Unter Windows 10 ist der Prozess ein bisschen einfacher. Um SonarQube Server zu starten ist die Ausführung der Windows-Eingabeaufforderung in windowsx86-64 in bin Ordner ist erforderlich.



Den Befehl StartSonar.bat in cmd eingeben und mit ein bisschen Geduld, ist SonarQube

Server Bereit:

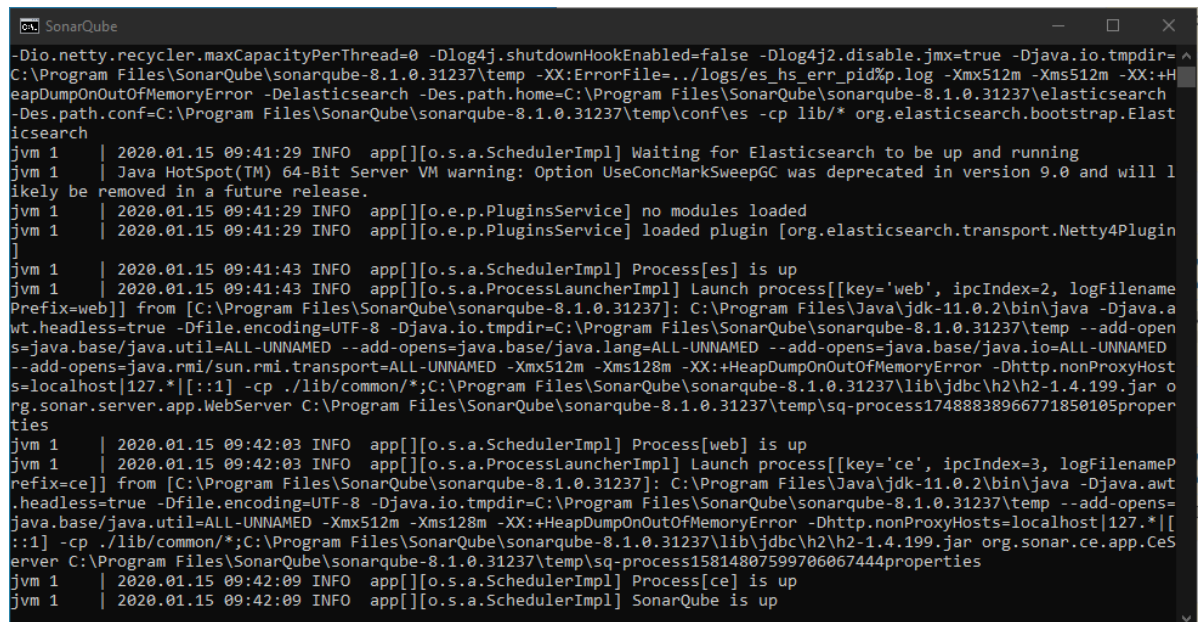


```
Microsoft Windows [Version 10.0.18363.535]
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\Mo\Downloads\sonarqube-6.7.7\sonarqube-6.7.7\bin\windows-x86-64>StartSonar.bat

wrapper | --> Wrapper Started as Console
wrapper | Launching a JVM...
jvm 1 | Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
jvm 1 | Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.
jvm 1 | 2020.01.14 11:18:28 INFO app[[o.s.a.AppFileSystem] Cleaning or creating temp directory C:\Users\Mo\Download
s\sonarqube-6.7.7\sonarqube-6.7.7\temp
jvm 1 | 2020.01.14 11:18:28 INFO app[[o.s.a.es.EsSettings] Elasticsearch listening on /127.0.0.1:9001
jvm 1 | 2020.01.14 11:18:28 INFO app[[o.s.a.p.ProcessLauncherImpl] Launch process[[key='es', ipcIndex=1, logFilenam
ePrefix=es]] from [C:\Users\Mo\Downloads\sonarqube-6.7.7\sonarqube-6.7.7\elasticsearch]: C:\Program Files\Java\jdk-11.0.
2\bin\java -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+AlwaysP
reTouch -server -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -Djdk.io.permissionsUseCanonicalP
ath=true -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dlog4
j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Dlog4j.skipJansi=true -Xms512m -Xmx512m -XX:+HeapDumpOnOutOfMemor
yError -Delasticsearch -Des.path.home=C:\Users\Mo\Downloads\sonarqube-6.7.7\sonarqube-6.7.7\elasticsearch -cp lib/* org.
elasticsearch.bootstrap.Elasticsearch -Epath.conf=C:\Users\Mo\Downloads\sonarqube-6.7.7\sonarqube-6.7.7\temp\conf\es
jvm 1 | 2020.01.14 11:18:28 INFO app[[o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
jvm 1 | Java HotSpot(TM) 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will l
ikely be removed in a future release.
jvm 1 | 2020.01.14 11:18:29 INFO app[[o.e.p.PluginsService] no modules loaded
jvm 1 | 2020.01.14 11:18:29 INFO app[[o.e.p.PluginsService] loaded plugin [org.elasticsearch.transport.Netty4Plugin
]
jvm 1 | WARNING: An illegal reflective access operation has occurred
jvm 1 | WARNING: Illegal reflective access by io.netty.util.internal.ReflectionUtil (file:/C:/Users/Mo/Downloads/sona
rqube-6.7.7/sonarqube-6.7.7/lib/common/netty-common-4.1.13.Final.jar) to constructor java.nio.DirectByteBuffer(long,int)
```

Wenn Sonar Server richtig gestartet ist dann sieht das cmd ungefähr so aus:



```
-Dio.netty.recycler.maxCapacityPerThread=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.io.tmpdir=C:\
Program Files\SonarQube\sonarqube-8.1.0.31237\temp -XX:ErrorFile=..logs/es_hs_err_pid%p.log -Xmx512m -Xms512m -XX:+H
eapDumpOnOutOfMemoryError -Delasticsearch -Des.path.home=C:\Program Files\SonarQube\sonarqube-8.1.0.31237\elasticsearch
-Des.path.conf=C:\Program Files\SonarQube\sonarqube-8.1.0.31237\temp\conf\es -cp lib/* org.elasticsearch.bootstrap.Elast
icsearch
jvm 1 | 2020.01.15 09:41:29 INFO app[[o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
jvm 1 | Java HotSpot(TM) 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will l
ikely be removed in a future release.
jvm 1 | 2020.01.15 09:41:29 INFO app[[o.e.p.PluginsService] no modules loaded
jvm 1 | 2020.01.15 09:41:29 INFO app[[o.e.p.PluginsService] loaded plugin [org.elasticsearch.transport.Netty4Plugin
]
jvm 1 | 2020.01.15 09:41:43 INFO app[[o.s.a.SchedulerImpl] Process[es] is up
jvm 1 | 2020.01.15 09:41:43 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFilename
Prefix=web]] from [C:\Program Files\SonarQube\sonarqube-8.1.0.31237]: C:\Program Files\Java\jdk-11.0.2\bin\java -Djava.a
wt.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Program Files\SonarQube\sonarqube-8.1.0.31237\temp --add-open
s=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED
--add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHost
s=localhost|127.*|::1 -cp ./lib/common/*;C:\Program Files\SonarQube\sonarqube-8.1.0.31237\lib\jdbc\h2\h2-1.4.199.jar o
rg.sonar.server.app.WebServer C:\Program Files\SonarQube\sonarqube-8.1.0.31237\temp\sq-process1748838966771850105proper
ties
jvm 1 | 2020.01.15 09:42:03 INFO app[[o.s.a.SchedulerImpl] Process[web] is up
jvm 1 | 2020.01.15 09:42:03 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenameP
refix=ce]] from [C:\Program Files\SonarQube\sonarqube-8.1.0.31237]: C:\Program Files\Java\jdk-11.0.2\bin\java -Djava.awt
.headless=true -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Program Files\SonarQube\sonarqube-8.1.0.31237\temp --add-opens=
java.base/java.util=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|
::1 -cp ./lib/common/*;C:\Program Files\SonarQube\sonarqube-8.1.0.31237\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ce.app.CeS
erver C:\Program Files\SonarQube\sonarqube-8.1.0.31237\temp\sq-process15814807599706067444properties
jvm 1 | 2020.01.15 09:42:09 INFO app[[o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2020.01.15 09:42:09 INFO app[[o.s.a.SchedulerImpl] SonarQube is up
```

Ausführen von SonarQube als Dienst unter Windows 10:

Installation oder Deinstallation des NT-Diensts könnten mit der Ausführung dieser Dateien als Administrator wirken:

%SONARQUBE_HOME%/bin/windows-x86-64/InstallNTService.bat

%SONARQUBE_HOME%/bin/windows-x86-64/UninstallNTService.bat

Und zum Starten oder Stoppen der Service:

%SONARQUBE_HOME%/bin/windows-x86-64/StartNTService.bat

%SONARQUBE_HOME%/bin/windows-x86-64/StopNTService.bat

SonarQube Plug-ins:

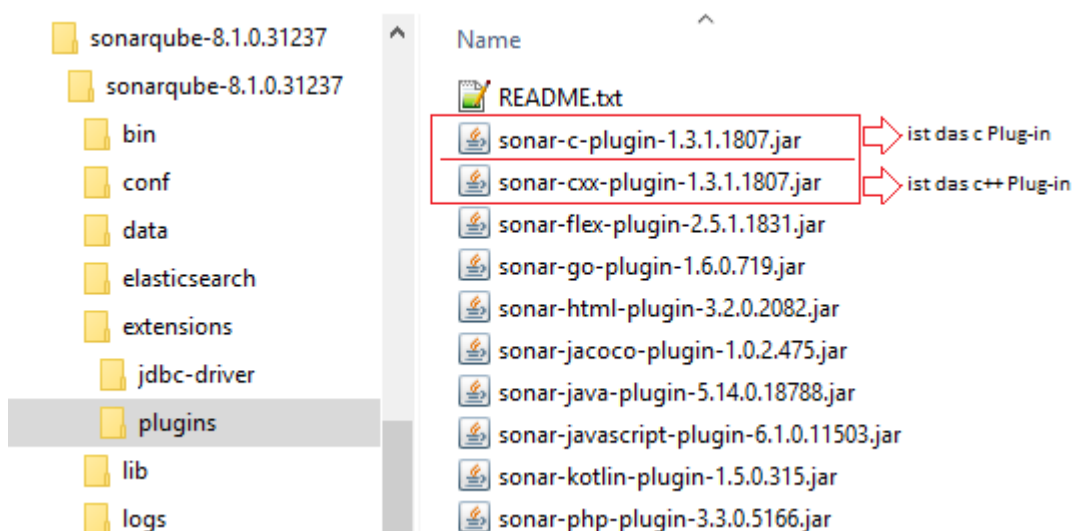
Es gibt zwei Möglichkeiten, ein Plugin in SonarQube zu installieren.

- Die erste ist als Administrator in Marketplace (Marktplatz), wo die Teile der SonarQube-Plattform auf dem neuesten Stand gehalten werden.

In Marktplatz erscheinen verschiedene Abschnitte: diese zeigt die installierte SonarQube Edition, welche Plugins sind installiert, und ob Updates verfügbar und kompatibel sind.

- Oder die Manuelle Installation, wo die Plugins herunterlädt und die heruntergeladene JAR-Datei in SonarQube Ordner / extensions /plugins platziert werden.

Hinweis: Die Entfernung aller vorherigen Versionen derselben Plugins ist nötig. Nach jeder neuen Plugins Installation oder Update muss der Server neugestartet werden.



Ähnlich wie bei anderen SonarQube-Plugins implementiert sonar-cxx einen internen Analysator, den sogenannten "SQUID Sensor". Die Importfunktionalität für externe Tools wird ebenfalls mit entsprechenden Sensoren implementiert.

Sensoren zur statischen und dynamischen Code-Analyse, die von diesem Plugin benutzt werden:

- Cppcheck Warnungen Unterstützung
- GCC/G++ Warnungen Unterstützung
- Visual Studio Warnungen Unterstützung
- Visual Studio Überprüfungen für C++ Core Guidelines
- Clang Static Analyzer Unterstützung
- Clang Tidy Warnungen Unterstützung
- PC-Lint Warnungen Unterstützung
- RATS
- Valgrind
- Vera++
- Dr. Memory Warnungen Unterstützung

Jede Art von Warnung, die von einem Sensor ausgegeben werden kann, wird durch eine Qualitätsregel dargestellt. Regeln werden in Regel Repositories gruppiert, wobei eine Beziehung zwischen einem Sonar-cxx-Sensor und dem Regel Repositories besteht. z.B. sonar-cxx installiert die folgenden Repositories:

PC-lint, Compiler-VC, Cppcheck, RATS, Compiler-GCC, Clang-Tidy, Clang-SA, Vera++, Valgrind, und C++ SonarQube (SQUID).

Alle Regeln für eine bestimmte Sprache werden in Qualitätsprofile gruppiert. z.B. Für die Sprache "C++ (Community)" installiert sonar-cxx das Standardqualitätsprofil "Sonar Way".

Analyse ausführen mit Sonar-Scanner:

Überblick:

Nach der Installation der SonarQube-Plattform sollte ein Scanner installiert und Projekte erstellt werden. Dazu muss man je nach Anforderungen den Scanner konfigurieren.

Durch die Unterstützung von Apaches Maven und Ant, MSBuild, Gradle und Jenkins erschließen sich vielfältige und einfache Möglichkeiten, die Analyse in den CI-Prozess zu integrieren. Die Analyse kann aber auch auf der lokalen Entwicklungsmaschine ausgeführt werden.

Diese verschiedenen Konfigurationen sind zu Verfügung:

- Sonar-Scanner für Gradle
- Sonar-Scanner für MSBuild
- Sonar-Scanner für Maven
- Sonar-Scanner für Jenkins
- Sonar-Scanner für
- Sonar-Scanner für Ant
- Sonst CLI (Kommandozeile)–Sonar-Scanner, was für das Projekt benötigt war.

CLI Sonar-Scanner:

Erfordert eine vorinstallierte JVM und hat denselben Anforderungen wie der SonarQube-Server.

Allererstes sollte man die benötigten Dateien herunterladen. Hier sind die Links für [Windows 64-bit](#) und [Linux 64-bit](#).

Danach sollte die Zip Datei ausgepackt werden wie bei der Installation von SonarQube.

In dem ausgepackten Ordner findet man die Datei `/conf/sonar-scanner.properties`, wo die globalen Einstellungen (host, Server ...) aktualisiert werden können:

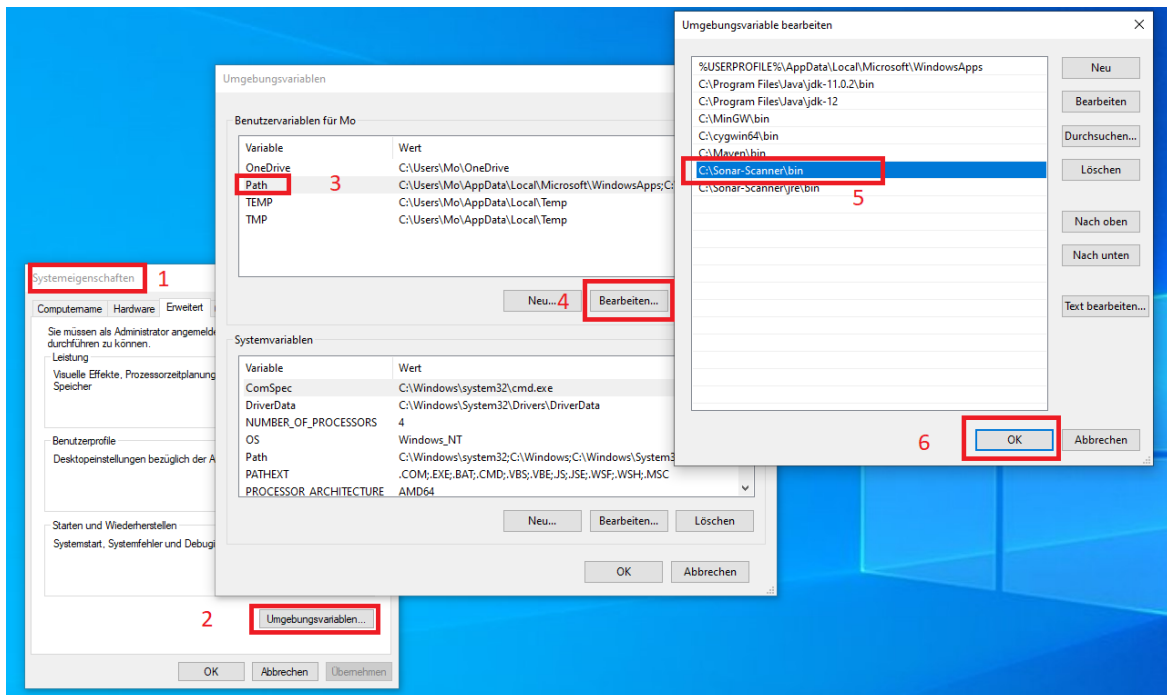

```
#Configure here general information about the scanner
#No information about specific project :

#----- Default SonarQube server
#sonar.host.url=http://localhost:9000

#----- Default source code encoding
#sonar.sourceEncoding=UTF-8
```

Es ist wichtig, dass, Sonar-Scanner bin Verzeichnis zu Pfad (PATH) hinzugefügt wird.

Unter Windows 10:



Auf einem Computer mit Ubuntu Linux ist die Installation von Sonar-Scanner anders,

dies erfolgt die folgenden Schritte:

- Installation der erforderlichen Pakete mit dem Befehl: `apt-get install unzip wget`
- Das Sonar-Scanner Paket herunterladen und verschieben in das OPT-Verzeichnis:
 - `mkdir /downloads/sonarqube -p`
 - `cd /downloads/sonarqube`
 - `wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.2.0.1873-linux.zip`
 - `unzip sonar-scanner-cli-4.2.0.1873-linux.zip`
 - `mv sonar-scanner-xxx-linux /opt/sonar-scanner` „xxx ist die Version“

- Bearbeitung die Datei sonar-scanner.properties ist wie bei Windows aber mit nano, vi usw..
- Auch hier soll den Sonar-Scanner-Befehl zur PATH-Variablen hinzugefügt,

und das erfolgt die Erstellung eine sh (Shell) Datei, um die erforderliche Konfiguration

der Umgebungsvariablen zu automatisieren:

Die Datei Erstellung mit dem Befehl: vi /etc/profile.d/sonar-scanner.sh Hier ist der Inhalt der sonar-scanner.sh-Datei:

```
#!/bin/bash
export PATH="$PATH:/opt/sonar-scanner/bin"
```

Anschließend, den Computer neustarten, und danach den Befehl source /etc/profile.d/sonar-scanner.sh, um den Befehl sonar scanner zur Variablen PATH hinzuzufügen.

```
C:\Users\Mo\Desktop\sonar-scanner>sonar-scanner -h
INFO:
INFO: usage: sonar-scanner [options]
INFO:
INFO: Options:
INFO:  -D,--define <arg>      Define property
INFO:  -h,--help              Display help information
INFO:  -v,--version            Display version information
INFO:  -X,--debug              Produce execution debug output
```

Überprüfung der Installation, bei der Öffnung eine neue Shell und mit der Ausführung des Befehls sonar-scanner -h (sonar-scanner.bat -h unter Windows). Es sollte die Ausgabe folgendermaßen aussehen:

Sonar-Scanner Verwendung:

Sonar-Scanner wird mit dem Befehl sonar-scanner in cmd im Projektbasisverzeichnis ausgeführt.

Aber davor soll die Konfigurationen von sonar-scanner definiert werden Entweder Bei der Erstellung einer Konfigurationsdatei (sonar-project.properties) im Stammverzeichnis des Projektes oder direkt im Kommandozeilen. z.B.:

- `sonar-scanner -D"sonar.projectKey=myproject" -D"sonar.sources=src1"`

Die Analyseparameter sind unter den Link auf der offiziellen Website [„Analysis Parameters“](#)

Bei manchen Plug-ins, wie z.B. Das C ++ Community Plugin Neben den allgemeinen SonarQubes Analyseparametern unterstützt das zuletzt veröffentlichte Plugin [die folgenden Parameter](#).

SonarQube Webinterface:

Wie bereits erwähnt, SonarQube stellt die Ergebnisse mithilfe einer Weboberfläche dar.

Folgendes wird die Weboberfläche kurz beschrieben:

Unter localhost:9000 im Browser, ist SonarQube Website Standardmäßig zu finden.

Der Einstiegspunkt für jedes Projekt ist die Projekt Homepage, die Folgendes anzeigt:

- der aktuelle Stand seiner Qualität
- die Qualität dessen, was seit Beginn der neuen Code-Periode produziert wurde.
- den Freigabestatus des Projekts

Das Quality Gate ist das Tool zur Durchsetzung von den erwarteten Qualitätsrichtlinien.

Die Seite beginnt mit dem aktuellen Quality Gate-Status des Projekts. das Projekt erfolgreich abgeschlossen wurde, wird eine einfache, grüne Entwarnung angezeigt. Ist dies nicht der Fall, sind Details und Drilldowns sofort verfügbar, um schnell zu identifizieren, was schiefgelaufen ist.