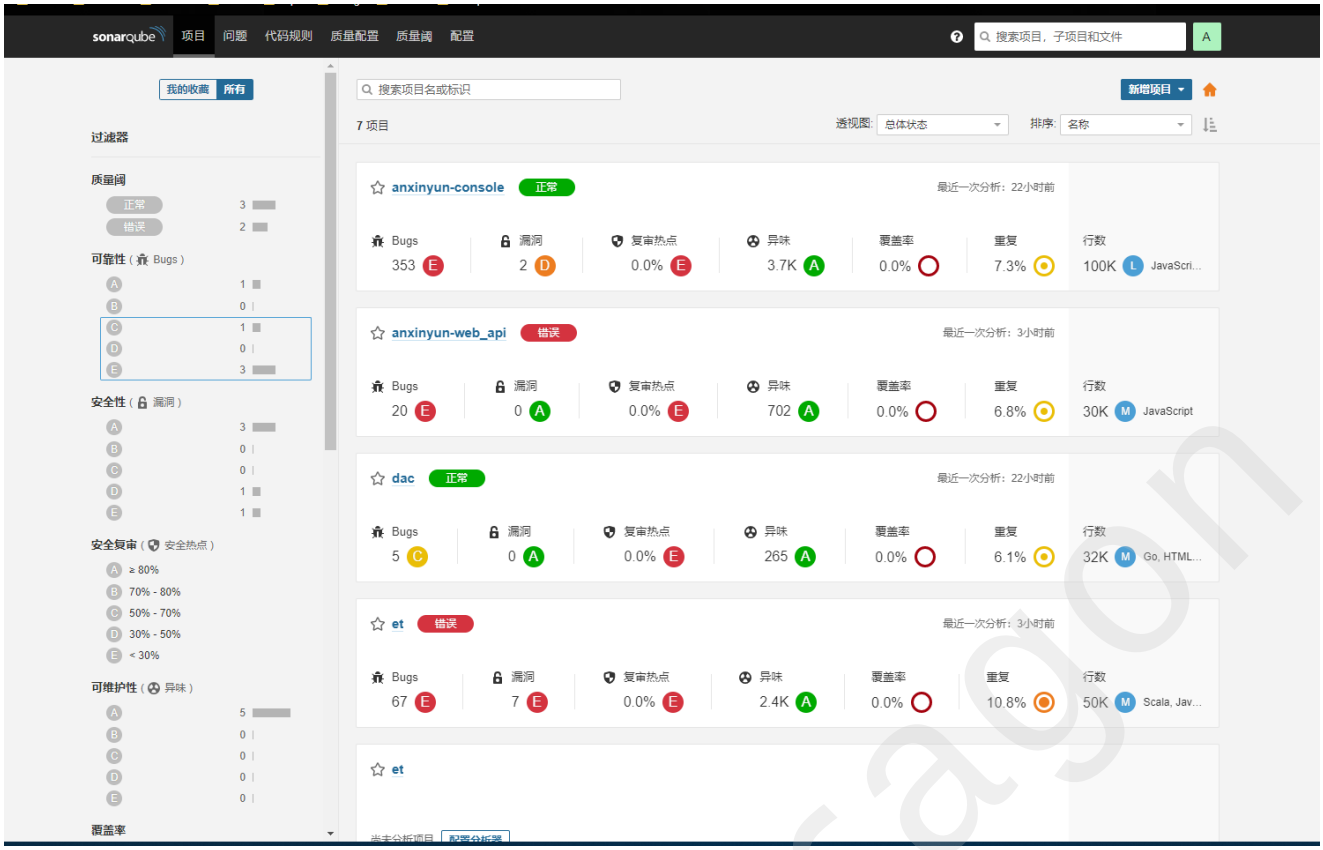
服务端部署

```yaml
version: "3"
services:
  sonarqube:
    image: sonarqube:lts-community
    environment:
      SONAR_JDBC_URL: jdbc:postgresql://10.8.30.156:5432/sonarqube
      SONAR_JDBC_USERNAME: sonarqubeAdmin
      SONAR_JDBC_PASSWORD: sonarqubeAdmin_123
      SONAR_JDBC_MAXACTIVE: "100"
      SONAR_JDBC_MAXIDLE: "10"
      SONAR_JDBC_MINIDLE: "5"
    volumes:
      - sonarqube_data:/opt/sonarqube/data
      - sonarqube_extensions:/opt/sonarqube/extensions
      - sonarqube_logs:/opt/sonarqube/logs
    ports:
      - "9000:9000"
volumes:
  sonarqube_data:
  sonarqube_extensions:
  sonarqube_logs:
```

部署成功：



体验

在服务端新增项目：

这个项目是一个maven 构建的项目，项目是scala 的，

按照提示执行 `maven`命令：

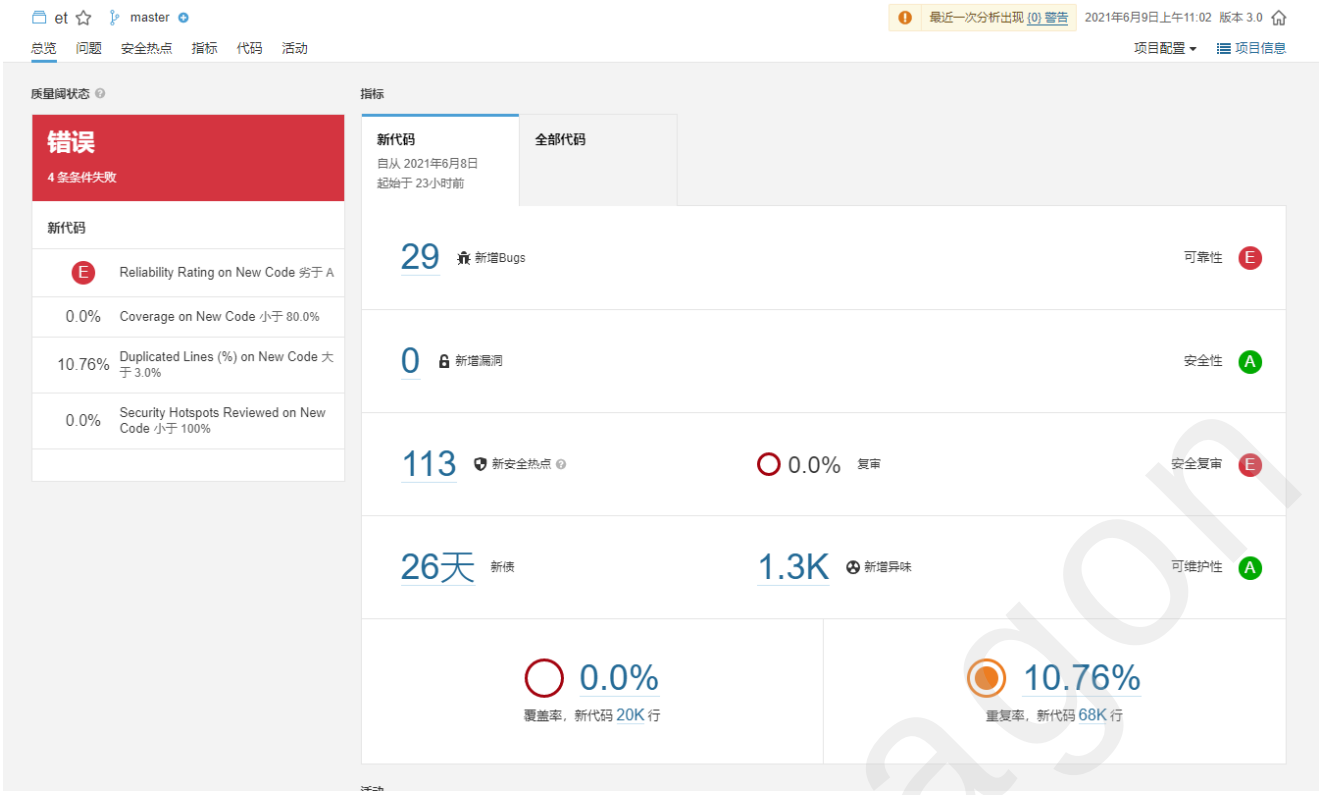
```
mvn sonar:sonar \
  -Dsonar.projectKey=et \
  -Dsonar.host.url=http://10.8.30.158:9000 \
  -Dsonar.login=d3ce7e00c8fcc9adc3b85b97d3c1005766cb2498
```

使用 `idea ide` 的 可以在安装插件，

> `File` ——> `Settings...` ——> `Plugins`
> 查找 `sonar` 可以找到插件，安装即可

安装 sonarscanner

执行后，代码扫描完成，

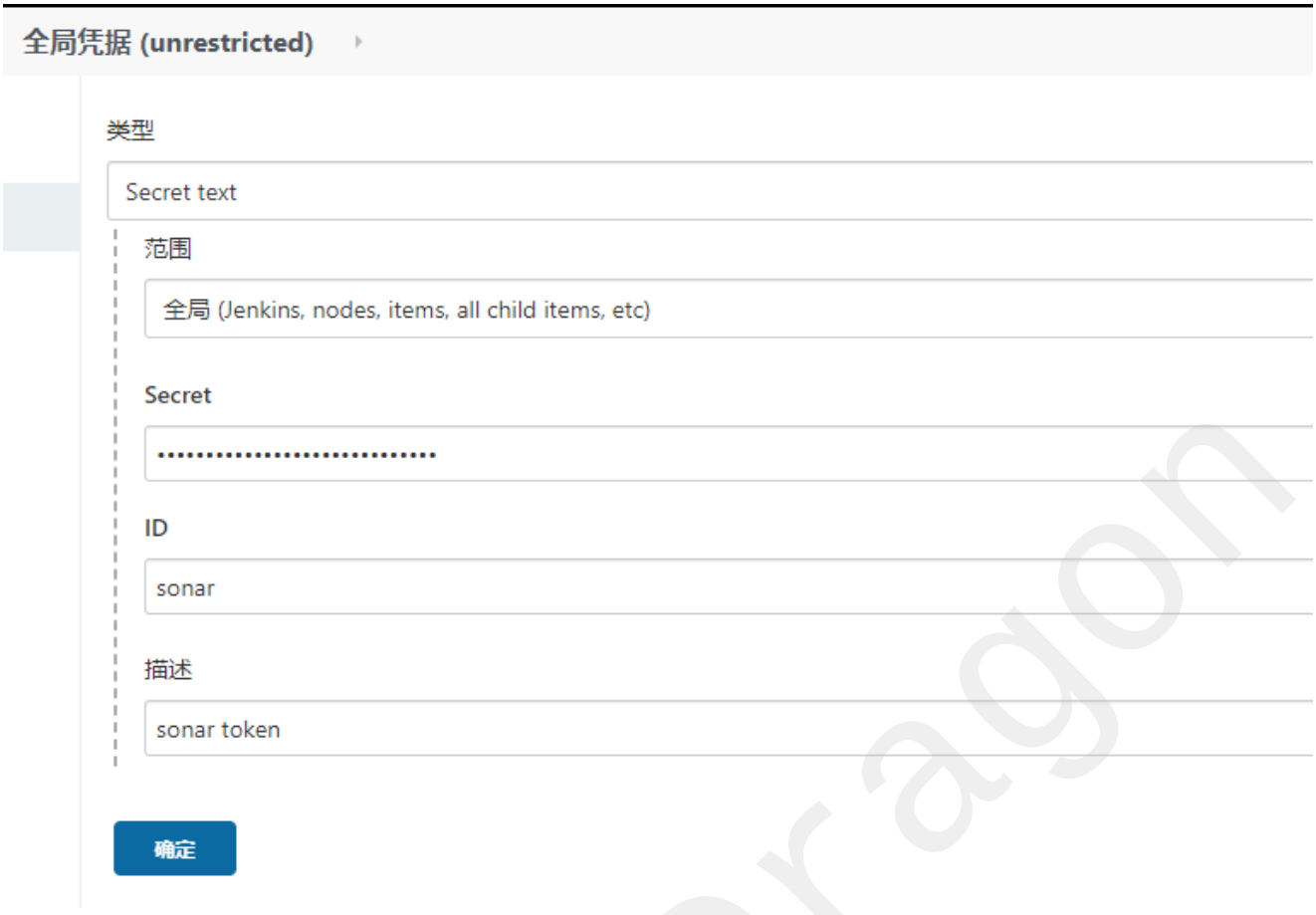


可以看到总览。

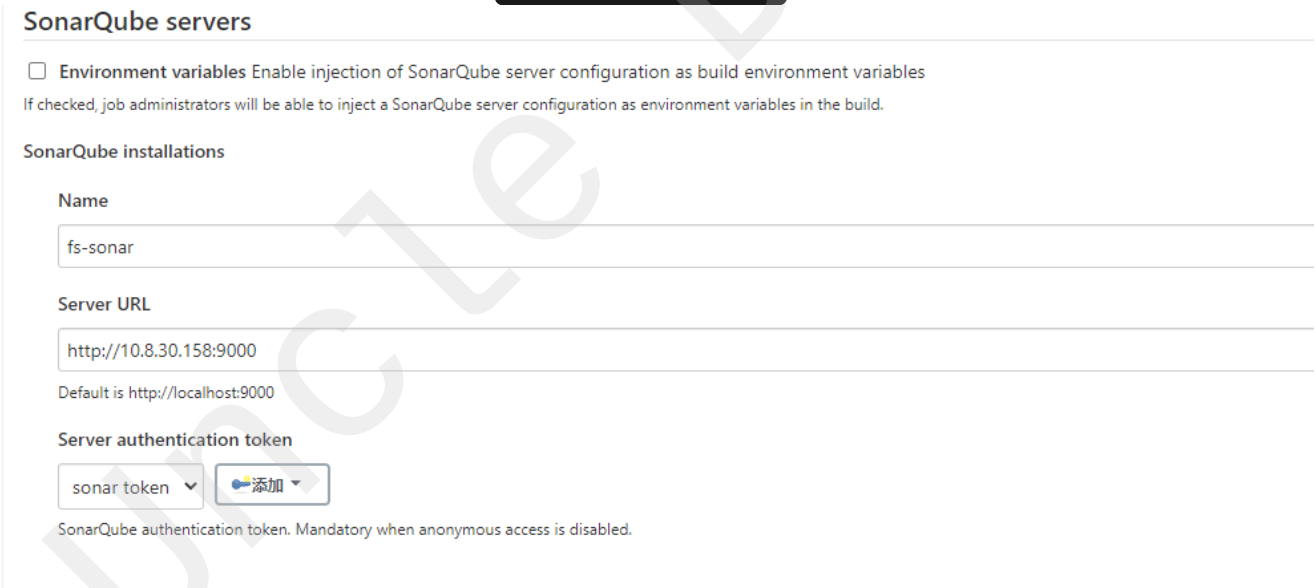
Jenkins 集成

Jenkins 安装 插件 `SonarQube Scanner for Jenkins`

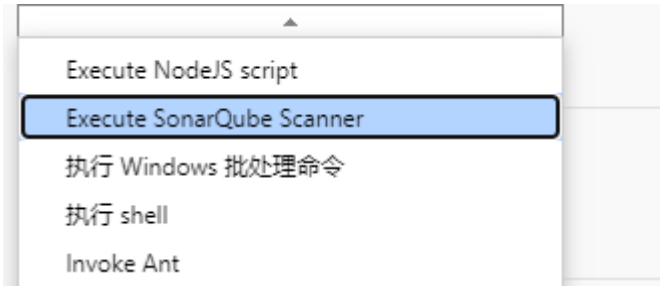
在 Jenkins 添加 凭据：



在 系统管理 ——> 系统配置中 ——`SonarQube servers`

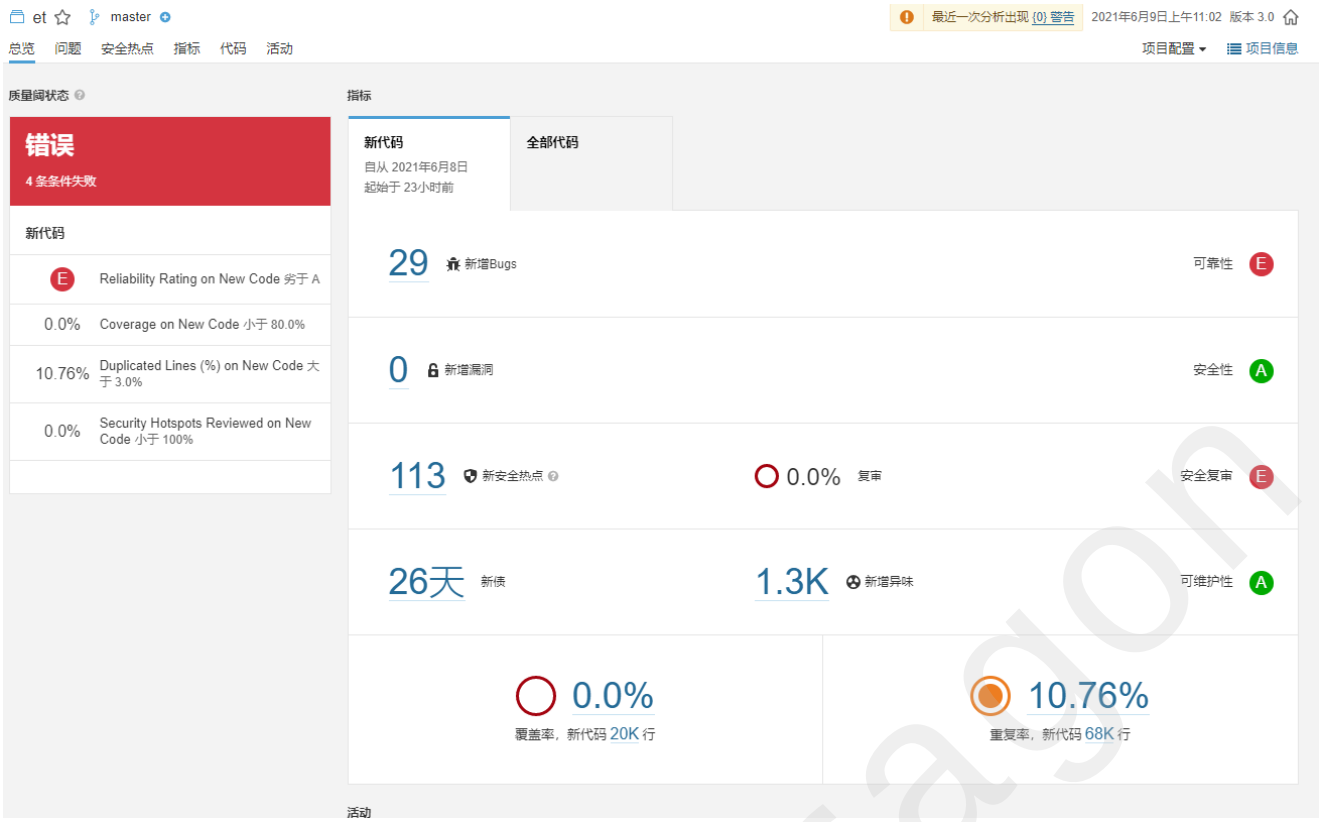


在项目构建配置中添加



构建成功完成后：

点击上图框的图标，就可以看到 静态扫描的总览了

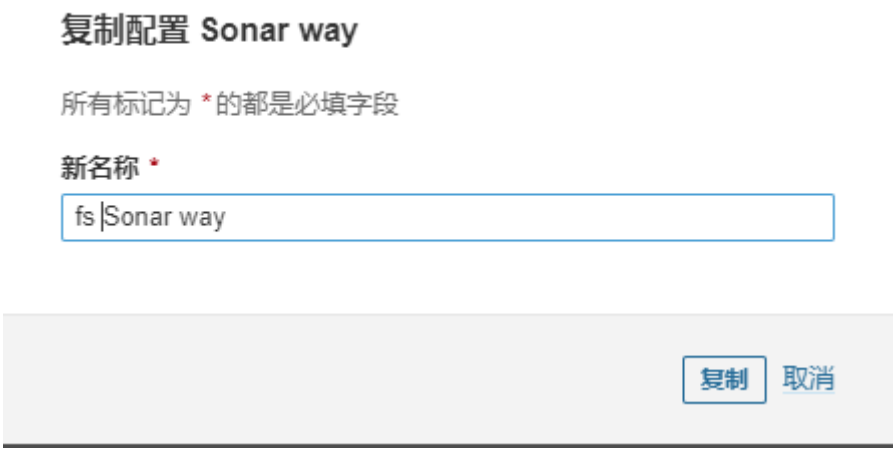


代码规则

`sonarQube` 支持自定义代码规则

下面以 go 语言配置为例：

内置的规则不能直接修改，复制内置的规则模板：



可以选择 启用和禁用相关规则：

"=+" should not be used instead of "+="	Go Bug	挂起
"switch" statements should not have too many "case" clauses	Go 异味 brain-overload	挂起
All branches in a conditional structure should not have exactly the same implementation	Go Bug	挂起
All code should be reachable	Go Bug cwe, unused	挂起
Boolean checks should not be inverted	Go 异味 pitfall	挂起
Boolean literals should not be redundant	Go 异味 clumsy	挂起
Cognitive Complexity of functions should not be too high	Go 异味 brain-overload	挂起
Function and method names should comply with a naming convention	Go 异味 convention	挂起
Functions should not be empty	Go 异味 suspicious	挂起
Functions should not have identical implementations	Go 异味 confusing, duplicate, suspicious	挂起
Functions should not have too many parameters	Go 异味 brain-overload	挂起
Hard-coded credentials are security-sensitive	Go 安全热点 cwe, owasp-a2, sans-top25-porous	挂起
Identical expressions should not be used on both sides of a binary operator	Go Bug	挂起
Local variable and function parameter names should comply with a naming convention	Go 异味 convention	挂起
Multi-line comments should not be empty	Go 异味	挂起
Nested blocks of code should not be left empty	Go 异味 suspicious	挂起