Instrukcja podłączenia serwera SonarQube do asystenta AI przez MCP, żeby asystent mógł analizować kod regułami naszego serwera.

> Ta instrukcja dotyczy wyłącznie asystenta omp (oh-my-pi). Claude Code, Cursor, Codex i inne narzędzia mają własne pliki konfiguracyjne o innym formacie i ta instrukcja ich nie obejmuje.

MCP (Model Context Protocol) to protokół, przez który asystent korzysta z zewnętrznych narzędzi. Serwer MCP SonarQube udostępnia asystentowi narzędzia do analizy kodu, przeglądania zgłoszeń i stanu bramki jakości na naszym serwerze. Działa w kontenerze Dockera, który omp uruchamia automatycznie przy starcie sesji.

Wymagania

- Docker zainstalowany i dostępny dla Twojego użytkownika bez sudo (polecenie `docker run hello-world` działa).
- Konto na serwerze sonar.suncode.local z dostępem do projektu PlusWorkflow.
- Certyfikat naszego lokalnego urzędu certyfikacji (Caddy) w pliku `/usr/local/share/ca-certificates/caddy-local-root-2023.crt`. Jeśli przeglądarka otwiera Sonara bez ostrzeżenia o certyfikacie, plik zwykle już jest na miejscu.

Krok 1. Wygeneruj token

W Sonarze otwórz My Account → Security, wpisz nazwę tokena (np. `omp-mcp`), wybierz typ User Token i kliknij Generate. Skopiuj token — Sonar pokazuje go tylko raz.

Krok 2. Dodaj serwer do konfiguracji omp

Otwórz plik `~/.omp/agent/mcp.json`. Jeśli plik nie istnieje, utwórz go z poniższą zawartością. Jeśli istnieje, dopisz sam wpis `"sonarqube"` do istniejącej sekcji `"mcpServers"`. W miejsce `TWOJ_TOKEN` wstaw token z kroku 1.

```
{
  "mcpServers": {
    "sonarqube": {
      "type": "stdio",
      "command": "docker",
      "args": [
        "run", "-i", "--rm", "--init", "--pull=always",
        "-e", "SONARQUBE_TOKEN",
        "-e", "SONARQUBE_URL",
        "--add-host", "sonar.suncode.local:192.168.1.99",
        "-v",
"/usr/local/share/ca-certificates/caddy-local-root-2023.crt:/u
sr/local/share/ca-certificates/caddy-local-root-2023.crt:ro",
        "mcp/sonarqube"
      ],
      "env": {
        "SONARQUBE_TOKEN": "TWOJ_TOKEN",
        "SONARQUBE_URL": "https://sonar.suncode.local"
      }
    }
  }
}
```

> Nie zapisuj tej konfiguracji w repozytorium (`.omp/mcp.json` w katalogu projektu), bo zawiera Twój prywatny token.

Co oznaczają mniej oczywiste elementy:

- `--add-host sonar.suncode.local:192.168.1.99` — kontener Dockera używa publicznych serwerów DNS i nie zna nazwy `sonar.suncode.local`. Ten wpis podaje mu adres serwera wprost.
- `-v …caddy-local-root-2023.crt…` — udostępnia kontenerowi certyfikat naszego urzędu certyfikacji. Bez niego kontener odrzuca połączenie z błędem `certificate verify failed`. Obraz `mcp/sonarqube` przy starcie instaluje certyfikaty z tego katalogu.
- `--pull=always` — przy każdym starcie pobiera najnowszą wersję obrazu serwera MCP.

Krok 3. Sprawdź połączenie

1. Uruchom nową sesję omp albo w trwającej sesji wpisz `/mcp reload`.
2. Poproś asystenta: „Sprawdź stan bramki jakości projektu PlusWorkflow w Sonarze”.
3. Asystent powinien zwrócić listę warunków bramki jakości z wartościami. Jeśli zamiast tego napisze, że nie ma narzędzi SonarQube, przejdź do sekcji „Rozwiązywanie problemów”.

Rozwiązywanie problemów

Objaw	Przyczyna	Co zrobić
Asystent nie ma narzędzi SonarQube	Błąd w pliku `mcp.json` albo Docker nie działa	Sprawdź, czy plik jest poprawnym JSON-em, i uruchom `docke hello-world`.
Błąd `certificate verify failed`	Brak certyfikatu Caddy pod podaną ścieżką	Sprawdź, czy plik `/usr/local/share/ca-certificates/ -root-2023.crt` istnieje. Jeśli masz go w innym miejscu, po po lewej stronie dwukropka w parametrze `-v`.
Błąd rozwiązywania nazwy albo przekroczony czas połączenia	Serwer Sonara zmienił adres IP	Sprawdź adres poleceniem `getent hosts sonar.suncode` popraw go w parametrze `--add-host`.
Błąd autoryzacji (401)	Nieprawidłowy albo unieważniony token	Wygeneruj nowy token (krok 1) i podmień go w pliku.

> Konfigurację sprawdziliśmy na Linuksie. Na Windows i macOS ścieżka certyfikatu jest inna, a sposób jego udostępnienia kontenerowi nie był testowany.