**Project Setup and Workflow**

**Phase 1: Local Setup (Project Hub)**

* **Goal:** Set up the foundational environment for data collection and code management.
* **Tasks:**
  1. **Clone Repositories**: Clone the open-source projects (e.g., Spring Boot, Apache Commons, JUnit 5) to your local machine.
  2. **Install SonarQube & Scanner**: Ensure SonarQube Community Edition and the SonarScanner CLI are installed and running locally.
  3. **Run Scans**: Execute a SonarScanner analysis on each repository to perform static analysis and send results to your local SonarQube server.

**Phase 2: Data Export & Storage**

* **Goal:** Transfer the raw, unfiltered code smell data to a cloud storage location.
* **Tasks:**
  1. **Use SonarQube Web API**: Use a Python script to interact with the SonarQube Web API. The api/issues/search endpoint is specifically designed to retrieve all issues in a project.
  2. **Directly Export to Google Drive**: Configure your script to directly export the JSON output from the API to your mounted Google Drive directory. This eliminates the need for a local download/re-upload step. The API can be used to get all raw data, which can then be formatted and displayed elsewhere.

**Phase 3: Google Colab (ML Lab)**

* **Goal:** Perform all data-intensive and computationally heavy tasks using cloud resources.
* **Tasks:**
  1. **Mount Google Drive**: In your Colab notebook, mount your Google Drive to seamlessly access the exported JSON files.
  2. **Data Preprocessing**: Load the raw data and prepare it for model training. This includes cleaning, normalization, and feature extraction.
  3. **Ground Truth Annotation**: Manually review a subset of the issues and apply your defined "true positive" and "false positive" labels.
  4. **Feature Engineering**: Extract features from the code snippets and project history, such as code embeddings and file churn.
  5. **Model Training**: Use the labeled data to train your ML-based post-processor classifier, leveraging Colab's GPU or TPU runtimes.
  6. **Evaluation**: Evaluate the model's performance on a held-out test set to measure metrics like precision and false positive reduction.

**Can I Dump SonarQube Data Directly to the Drive?**

Yes, you can absolutely do this. While SonarQube itself does not have a native, built-in feature to export directly to a cloud drive like Google Drive, you can easily automate this process.

The most effective way is to use a simple script that acts as a middleman:

1. **Run your local SonarQube scan** as you've planned.
2. **Use a Python script** that calls SonarQube's Web API to get the analysis results. You'll use an API endpoint like /api/issues/search to retrieve the data in JSON format.
3. **Configure the script to dump the data directly to Google Drive**. Since you have Google Drive's desktop app installed and synced, your script can simply write the JSON file to that directory, and the app will automatically handle the upload to the cloud.