AI Governance & Cybersecurity Labs — Getting Started Guide

This guide will help you set up your environment to run the hands-on labs for AI Governance & Cybersecurity. You can run the labs in three ways:  
1. Directly with Python  
2. Using Docker containers (recommended for safety & consistency)  
3. Using GitHub Codespaces (dev container) or GitHub Actions (CI/CD scans)

# 1. Running Labs Directly with Python

Steps:

* Install Python 3.10+ (3.11 preferred). Verify with: python3 --version
* Install pip (Python package manager).
* Install required packages: pip install flask requests beautifulsoup4
* Navigate to a lab folder and run it directly, e.g.: python lab1\_prompt\_injection/exploit.py

# 2. Running Labs with Docker (Recommended)

* Install Docker (v20+) and Docker Compose (v2). Verify with: docker --version and docker compose version
* Navigate to a lab folder (e.g., lab2\_ai\_generated\_code\_vulns).
* Build and run vulnerable + defended apps with: docker compose up --build
* In another terminal, run exploits (e.g., python exploit\_calc.py).

# 3. Running Labs with GitHub Codespaces (Dev Container)

* Push the repo to your own GitHub account.
* In GitHub, click Code → Codespaces → Create codespace on main.
* Wait for the dev container to build; dependencies (Flask, Requests, BeautifulSoup, Semgrep, Bandit) are pre-installed automatically.
* Ports 5001, 5002, 7001, 7002, 8088 are forwarded automatically for you.
* Run a lab inside Codespaces just like local Python: e.g., cd lab2\_ai\_generated\_code\_vulns; python seed\_db.py; FLASK\_APP=app\_vulnerable.py flask run -p 5001

# 4. Running with GitHub Actions (CI/CD)

* Push the lab repository to your GitHub account.
* GitHub Actions will automatically run Semgrep (static analysis) and Bandit (security linter).
* Check results under the 'Actions' tab in your repo.
* You can also run locally inside Codespaces or locally installed environment: semgrep scan --config lab2\_ai\_generated\_code\_vulns/semgrep\_rules.yml and bandit -r .

# 5. Optional Developer Tools

* Postman or curl for testing HTTP endpoints.
* VS Code, Cursor, or PyCharm for editing/debugging.
* Pre-commit hooks (black, ruff, bandit) for linting/security checks before commits.

# Final Notes

All secrets in the labs are FAKE — safe for demo use. Labs are for educational purposes only. Always run in a controlled environment.