National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Srikanth Konka
Student ID: x23245824

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Srikanth Konka
Student ID:	x23245824
Programme:	MSc Cloud Computing **Year:** 2024-25
Module:	MSc Research Project
Lecturer:	Sean Heeney
Submission Due Date:	24 April 2025
Project Title:	Detecting and Mitigating AWS-Specific Code Smells in Ansible Infrastructure as Code
Word Count: 1332	**Page Count:** 10

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Srikanth Konka

Date: 24/04/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Srikanth Konka
Student ID: x23245824

1 Introduction

This project configuration manual contains instructions to help the user setup the research project infrastructure for the AWS-Ansible Code Smell Detection Framework. This framework is intended to detect and analyze code smells in ansible IaC scripts written for AWS-focused Ansible infrastructure as code (IaC) scripts. This work follows a combined approach involving program dependence graph (PDG) analysis and multi-layer perceptron (MLP) neural networks to locate the different types of code smells present in Ansible playbooks and propose remediation strategies accordingly.

2 System Requirements

2.1 Local Hardware Requirements

- Processor: Intel Quad-core Processor or better
- RAM: 8GB Minimum, Recommended 16GB
- Storage: SSD Preferred with 256GB or more

2.2 Software Requirements

- OS: Linux (Ubuntu 20.04 LTS or newer), macOS, or Windows 10/11
- Python: 3.8 or later
- Python package manager: pip3
- Git: for repository cloning

2.2 Ansible Control Node Specifications

- OS: Linux (Ubuntu 20.04 LTS or newer)
- EC2 Instances: t2.medium or higher
- RAM: 4GB Minimum
- Storage: 24 GB minimum

3 AWS Setup Instructions

3.1 AWS IAM User Permission

- Login using IAM in AWS Console and create programmatic access user with these policies:
    - - AmazonEC2FullAccess
    - - AmazonVPCFullAccess

- - AmazonRDSFullAccess
- - AmazonS3FullAccess

- Save the access key ID and secret access key

3.2 Ansible Control Node Setup

- SSH into control node: ssh -i ansible-key.pem ubuntu@<control-node-ip>
- Update and install required packages
  - `sudo apt update`
  - `sudo apt upgrade -y`
  - `sudo apt install -y python3-pip git`
- Install Ansible and required Python packages
  - `pip3 install ansible boto3 botocore`
  - `pip3 install ansible-lint yamllint`
- Install Python machine learning packages for deep learning component
  - `pip3 install pandas scikit-learn matplotlib seaborn tensorflow`

3.3 AWS Credentials Configuration

```
# Configure AWS credentials
mkdir -p ~/.aws
cat > ~/.aws/credentials << EOF
[default]
aws_access_key_id = YOUR_ACCESS_KEY_ID
aws_secret_access_key = YOUR_SECRET_ACCESS_KEY
EOF

cat > ~/.aws/config << EOF
[default]
region = eu-west-1
output = json
EOF
```

4 Project Structure

- create a main project directory structure pattern as below:

```
bash
>> mkdir -p code-smell-detector
>> cd code-smell-detector
```

code-smell-detector

- combined/ #combined MLP-PDG Detector Tool
  - combined_smell_detector.py
  - ml-analyzer.py
  - pdg-analyzer.py
- ml/ #ML Analyser

        - features.csv
        - labels.csv
        - ml_analyzer.py
    - models/ # Models directory after training
        - smell-detector-model.h5
        - best_model.h5
        - feature_scaler.joblib
        - model_metadata.json
    - pdg/ #PDG Analyser
        - pdg_analyzer.py
    - playbooks/ #Ansible Playbooks Directory
        - deploy_vulnerable_instances.yml
        - deploy_vulnerable_security_groups.yml
        - deploy_vulnerable_webapp.yml
        - synthetic_HP1_main.yml
        - synthetic_HP1_site.yml
        - synthetic_SEC_main.yml
        - synthetic_SEC_site.yml
        - synthetic_UO1_main.yml
        - synthetic_UO1_site.yml
        - synthetic_UR1_main.yml
        - synthetic_UR1_site.yml
    - reports.html
    - results/ #Result save directory
        - refactoring/
        - deploy_vulnerable_instances_refactoring.md
        - deploy_vulnerable_security_groups_refactoring.md
        - deploy_vulnerable_webapp_refactoring.md
        - deploy_vulnerable_instances_report.html
        - deploy_vulnerable_security_groups_report.html
        - deploy_vulnerable_webapp_report.html
        - deploy_vulnerable_instances_analysis.json
        - deploy_vulnerable_security_groups_analysis.json
        - deploy_vulnerable_webapp_analysis.json
    - scripts/
    - generate_labels.py
    - training_data/
        - ansible-examples/ #GitHub Ansible Playbooks
        - ansible-for-deveops/ #GitHub Ansible Playbooks
        - playbooks2/ - #Synthetically generated Ansible Playbooks for ML Training

5 Ansible Configuration

```
# Create Ansible configuration
cat > ansible.cfg << EOF
[defaults]
inventory = inventory
host_key_checking = False
roles_path = roles
```

```
retry_files_enabled = False
stdout_callback = yaml

[inventory]
enable_plugins = aws_ec2
EOF

cat > inventory/aws_ec2.yml << EOF
plugin: aws_ec2
regions:
  - eu-west-1
keyed_groups:
  - key: tags.Environment
    prefix: env
  - key: tags.Role
    prefix: role
compose:
  ansible_host: public_ip_address
EOF
```

6 Test Ansible Playbook – vulnerable_environment.yml

```
# vulnerable_environment.yml
---
- name: Create AWS Test Environment with Intentional Code Smells
  hosts: localhost
  connection: local
  gather_facts: false
  vars:
    # Setup variables
    region: "eu-west-1"
    # UR1: Unsafe reuse due to impure initializer
    # Using random in a template expression that will be reused
    environment_id: "testenv-{{ 999 | random }}"
    vpc_cidr: "10.0.0.0/16"
    project_name: "ansible-smell-test"

  tasks:
    # VPC Creation
    - name: Create VPC
      amazon.aws.ec2_vpc_net:
        name: "{{ project_name }}-vpc"
        cidr_block: "{{ vpc_cidr }}"
        region: "{{ region }}"
        tags:
          Name: "{{ project_name }}-vpc"
          Environment: "{{ environment_id }}"
      register: vpc
```

```
        # UO1: Unconditional override - Overriding the environment_id
unnecessarily
    - name: Override environment ID unconditionally
      set_fact:
        environment_id: "testenv-override"

    # Create Subnets
    - name: Create public subnet
      amazon.aws.ec2_vpc_subnet:
        vpc_id: "{{ vpc.vpc.id }}"
        cidr_block: "10.0.1.0/24"
        map_public: yes
        region: "{{ region }}"
        tags:
          Name: "{{ project_name }}-public-subnet"
          Environment: "{{ environment_id }}"
      register: public_subnet

    # Create Private Subnet
    - name: Create private subnet
      amazon.aws.ec2_vpc_subnet:
        vpc_id: "{{ vpc.vpc.id }}"
        cidr_block: "10.0.2.0/24"
        region: "{{ region }}"
        map_public: no
        tags:
          Name: "{{ project_name }}-private-subnet"
          Environment: "{{ environment_id }}"
      register: private_subnet

    # Internet Gateway
    - name: Create Internet Gateway
      amazon.aws.ec2_vpc_igw:
        vpc_id: "{{ vpc.vpc.id }}"
        region: "{{ region }}"
        tags:
          Name: "{{ project_name }}-igw"
          Environment: "{{ environment_id }}"
      register: igw

    # Route Tables
    - name: Create public route table
      amazon.aws.ec2_vpc_route_table:
        vpc_id: "{{ vpc.vpc.id }}"
        region: "{{ region }}"
        subnets:
          - "{{ public_subnet.subnet.id }}"
        routes:
          - dest: 0.0.0.0/0
            gateway_id: "{{ igw.gateway_id }}"
        tags:
          Name: "{{ project_name }}-public-route-table"
          Environment: "{{ environment_id }}"
```

```
    # UO2: Unused override - Using vars block that won't take effect
    - name: Create security group with unused override
      amazon.aws.ec2_security_group:
        name: "web-server-sg"
        description: "Web server security group"
        vpc_id: "{{ vpc.vpc.id }}"
        region: "{{ region }}"
        rules:
          - proto: tcp
            ports: 80
            cidr_ip: 0.0.0.0/0
          - proto: tcp
            ports: 443
            cidr_ip: 0.0.0.0/0
          - proto: tcp
            ports: 22
            cidr_ip: 0.0.0.0/0  # Security smell - unrestricted SSH access
        vars:
              region: "eu-central-1"   # UO2: This will be ignored due to
variable precedence
        tags:
          Name: "{{ project_name }}-web-sg"
          Environment: "{{ environment_id }}"
      register: web_sg
```

7 Project Run Commands

7.1 PDG Analyzer

```
>>python3 pdg/pdg_analyzer.py
playbooks/deploy_vulnerable_security_groups.yml --verbose --visualize
>>python3 pdg/pdg_analyzer.py playbooks/deploy_vulnerable_instances.yml
--verbose --visualize
>>python3 pdg/pdg_analyzer.py playbooks/deploy_vulnerable_webapp.yml
--verbose --visualize
```

7.2 ML Analyzer

```
>>python3 ml/ml_analyzer.py extract playbooks/ --output ml/features.csv
>>python3 ml/ml_analyzer.py train ml/featuresv3.csv ml/labelsv3.csv
--output models
>>python3 ml/ml_analyzer.py analyze
playbooks/deploy_vulnerable_instances.yml --model models/best_model.h5
--scaler models/feature_scaler.joblib --metadata models/model_metadata.json
--output reports
>>python3 ml/ml_analyzer.py analyze
playbooks/deploy_vulnerable_security_groups.yml --model
models/best_model.h5 --scaler models/feature_scaler.joblib --metadata
models/model_metadata.json --output reports
>>python3 ml/ml_analyzer.py analyze playbooks/deploy_vulnerable_webapp.yml
--model models/best_model.h5 --scaler models/feature_scaler.joblib
--metadata models/model_metadata.json --output reports
```

7.3 Combined PDG-MLP Analyzer

```
>>python3 combined/combined_smell_detector.py analyze
playbooks/deploy_vulnerable_security_groups.yml --model
models/best_model.h5 --scaler models/feature_scaler.joblib --metadata
models/model_metadata.json --output results
>>python3 combined/combined_smell_detector.py analyze
playbooks/deploy_vulnerable_instances.yml --model models/best_model.h5
--scaler models/feature_scaler.joblib --metadata models/model_metadata.json
--output results
>>python3 combined/combined_smell_detector.py analyze
playbooks/deploy_vulnerable_webapp.yml --model models/best_model.h5
--scaler models/feature_scaler.joblib --metadata models/model_metadata.json
--output results
```

8 Combined PDG-ML Detection Tool - Output

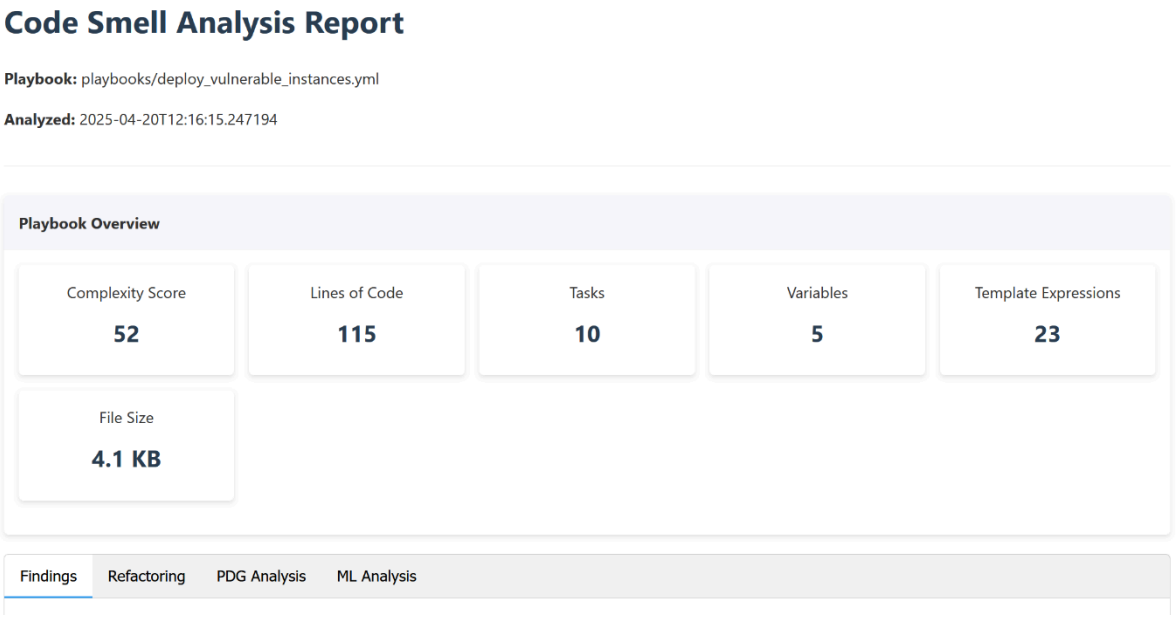


Figure 1. HTML Report – Dashboard

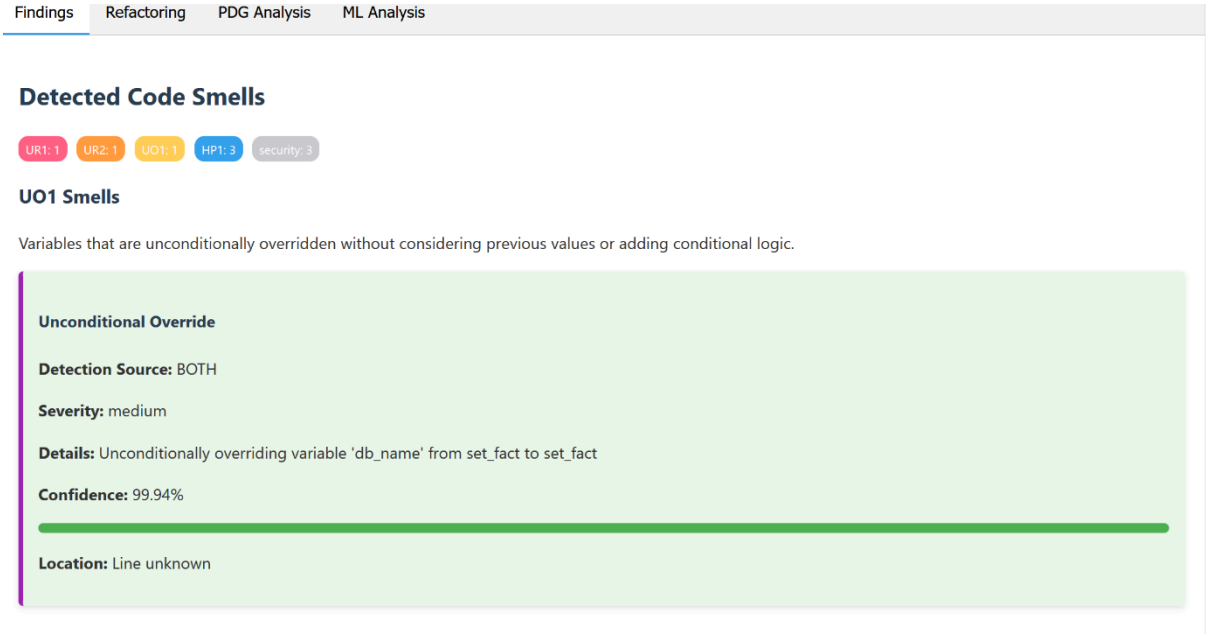
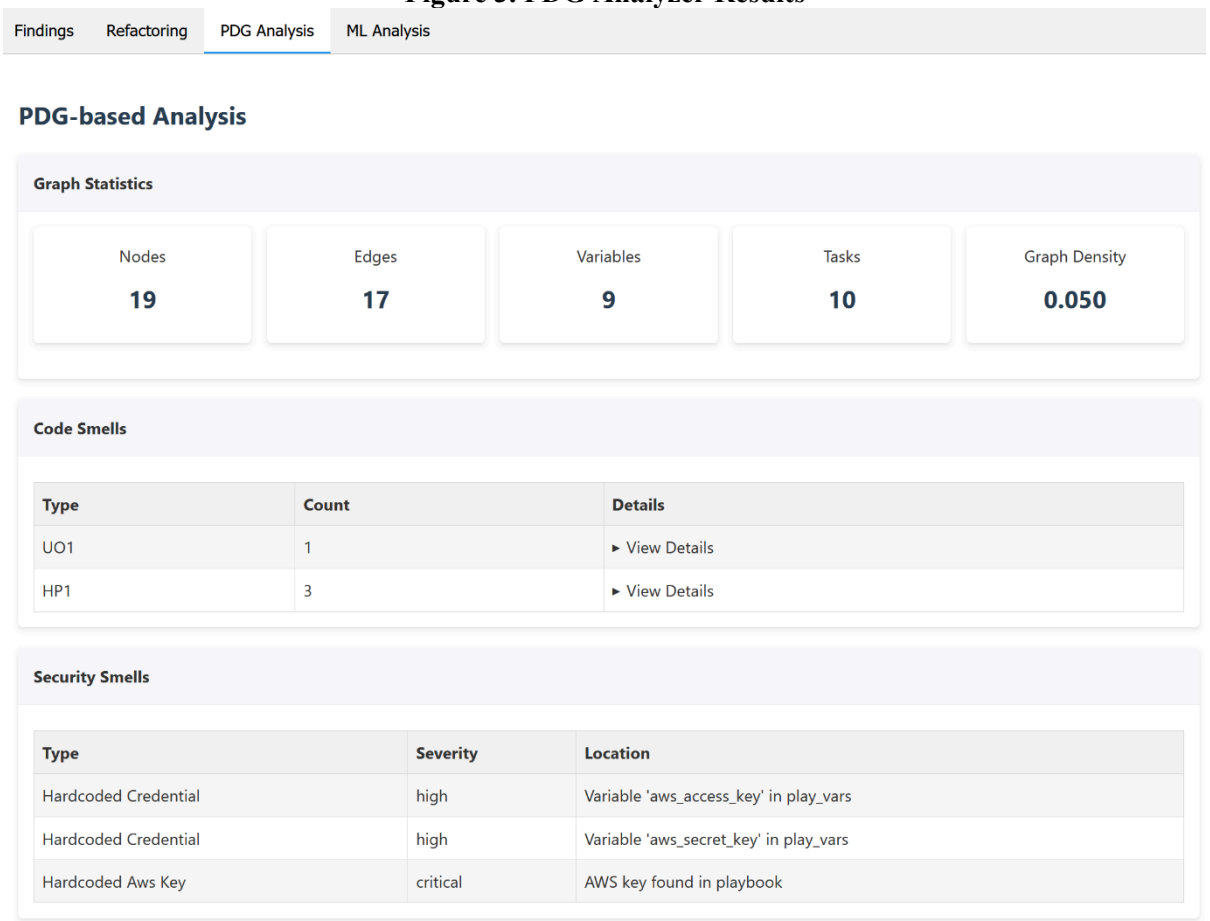


Figure 2. Detected Code Smells Window

Figure 3. PDG Analyzer Results



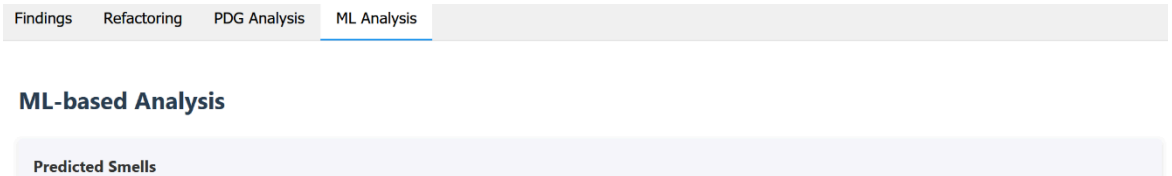


Figure 4. ML Analyzer Results

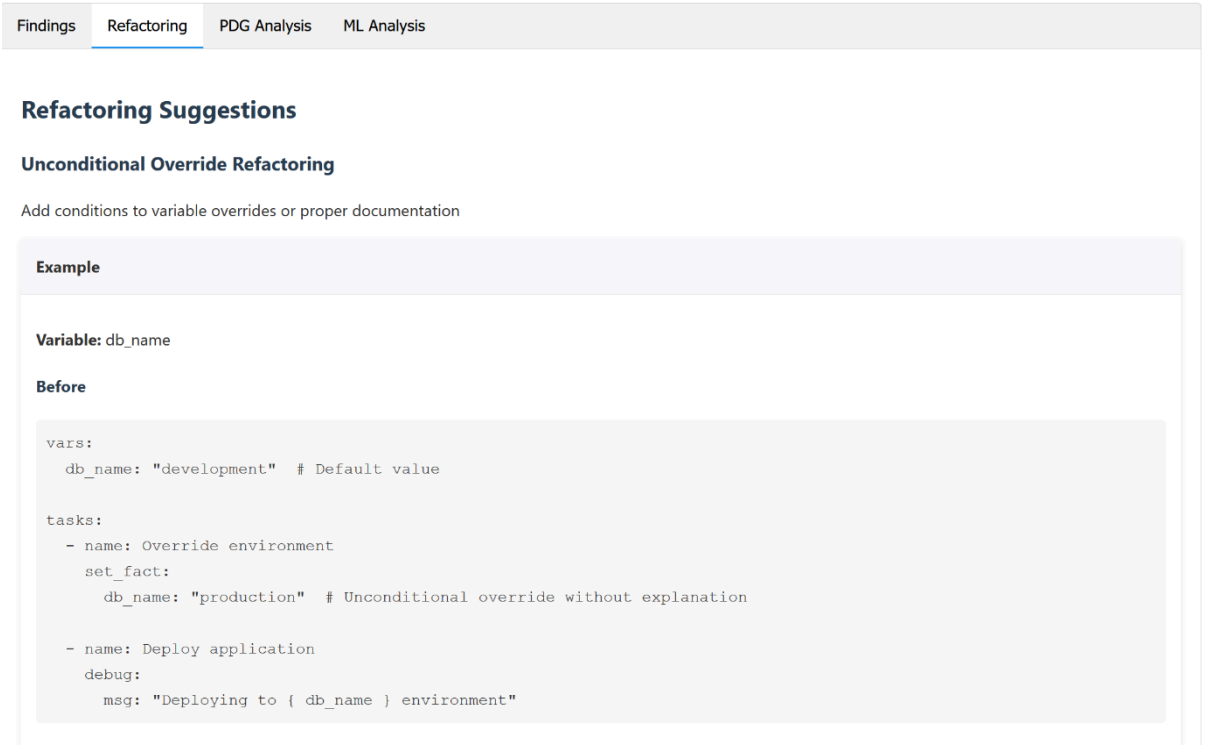


Figure 5. Refactoring Suggestions Window

References

Creating IAM users in your AWS account. Available at: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html (Accessed: 23 April 2025).

Getting started: Using Ubuntu. Available at: https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/gs-ubuntu.html (Accessed: 23 April 2025).

Automate Ansible playbook deployment with Amazon EC2 and GitHub. Available at: https://aws.amazon.com/blogs/infrastructure-and-automation/automate-ansible-playbook-deployment-amazon-ec2-github/ (Accessed: 23 April 2025).