TCS iON RIO Project Report

Docker-Based Lab Environment with Vulnerability Testing on AWS EC2

* **Introduction:**

This project was carried out as part of the TCS iON RIO internship initiative. It aims to create a secure and

scalable cloud-based lab environment using Docker on an AWS EC2 instance. The environment includes a

web server (Apache), a MySQL database, and basic vulnerability scanning using Nmap.

* **Objective :**

Set up a Docker-based environment on an Ubuntu EC2 instance.

Host a web application (Apache) and database (MySQL).

Establish a secure private network between containers.

Perform basic vulnerability scanning with Nmap.

* **Tools & Technologies Used:**

|  |  |
| --- | --- |
| TOOLS | PURPOSE |
| AWS EC2 | Cloud Hosting |
| Docker and Docker Compose | Container Management |
| Ubuntu 24.04 | Operating System |
| Apache(httpd) | Web Server |
| MySQL5.7 | Database |
| Nmap | Port Scanning |

* **Setup Steps:**

**4.1 AWS EC2 Instance**

OS: Ubuntu Server 24.04 LTS

Instance Type: t3.micro (Free Tier)

Security Group: Ports 22, 80, 3306 allowed

**4.2 Docker & Compose Installation**

sudo apt update && sudo apt upgrade -y

sudo apt install docker.io docker-compose -y

**4.3 Docker Compose Configuration**

Created a docker-compose.yml with:

• Apache on port 80 (172.16.0.100)

• MySQL on port 3306 (172.16.0.101)

• Static IPs and custom bridge network

**4.4 Run Services**

sudo docker-compose up -d

* **Testing:**
* Accessed Apache via: curl http://localhost
* Accessed MySQL using Docker CLI
* sudo docker exec -it mysql\_db mysql -uroot -p
* **Vulnerability Assessment :**

**Nmap Scan**

nmap -sV 172.20.0.0/24

• Found open ports: 22, 80, 3306

• Detected service versions

• Verified communication across subnet

* **Screenshots:**

Screenshots included:

• SSH to EC2 instance

• Docker containers running

• Apache and MySQL access

• Nmap scan results

* **Challenges & Learnings :**

• Learned to manage Docker networks and services.

• Understood EC2 networking and firewall configuration.

• Got hands-on with basic security testing using open-source tools.

* **Future Enhancements:**

• Add DVWA or Metasploitable for more complex attacks

• Use Nessus or OpenVAS for detailed vulnerability scans

• Secure MySQL with encryption and least-privilege users

* **Conclusion :**

The project helped build a practical understanding of Docker, AWS, and basic vulnerability assessment. It

provided insight into how enterprise systems host isolated environments with controlled networking and

security.