Configuration Manual

MSc Research Project
MSc in Cloud Computing

Alric Mendonsa
Student ID: 21227462

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Alric Mendonsa
Student ID:	21227462
Programme:	MSc in Cloud Computing
Year:	2024-2025
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	1066
Page Count:	5

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:	Alric Mendonsa
Date:	25th January 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

Alric Mendonsa
21227462

1 Prerequisites

- AWS account with administrative/root privileges
- EC2 Server or local system running Ubuntu 20.04 Operating System and above for managing Docker images
- Python version 3.8 and above alongwith virtual environment setup, pandas, boto3 and matplotlib libraries installed on EC2/local system
- AWS CLI configured on EC2 Server/local system to interact with AWS services
- Docker engine installed on EC2/local system
- Access to the dataset of publicly hosted Docker container images in the DockerHub repository

  `https://hub.docker.com/r/dh157/research_images_sept_2022`
  `https://hub.docker.com/r/dh157/research_images_dec_2022`
- Copy the buildspec.yml, scan-images.sh, scan-images-grype.sh files to EC2/local system and upload it to the main root directory of your GitHub repository as this will be required later when creating the CI/CD pipeline in AWS for the CodeBuild project and CodePipeline to connect to the GitHub repo and authenticate it using OAuth to trigger the pipeline execution.

2 Resource setup

2.1 Amazon Elastic Container Registry (ECR)

1. **Create an ECR Repository**:
   - Go to the AWS Management Console.
   - Navigate to **ECR** > **Repositories** > **Create Repository**.
   - Specify a name (e.g., `docker-security-assessment`).
   - Configure settings like scan on push and encryption.
2. **Push Docker Images**:

- Authenticate Docker with ECR:
  [] aws ecr get-login-password --region <region> — docker login --username AWS --password-stdin <account_id>.dkr.ecr.<region>.amazonaws.com
- Tag and push your Docker image:
  [] docker tag <image_name> <account_id>.dkr.ecr.<region>.amazonaws.com/<repository_r
  docker push <account_id>.dkr.ecr.<region>.amazonaws.com/<repository_name>

2.3 Amazon S3

1. **Create an S3 Bucket**:
   - Navigate to **S3** > **Create Bucket**.
   - Set the bucket name (e.g., `docker-security-results`) and region.
   - Configure permissions to allow secure access.
2. **Store Logs**:
   - Upload scan results and logs:
     [] aws s3 cp <file_path> s3://<bucket_name>/<key>

2.4 Amazon EC2

1. **Launch an Instance**:
   - Navigate to **EC2** > **Launch Instances**.
   - Choose an instance type (e.g., t2.medium for small workloads or t3.large/t3.xlarge for moderate to high level workloads).
   - Configure storage (20–50GB depending on requirements).
   - Assign a security group allowing SSH (port 22) and other necessary ports.
2. **Install Docker**:
   - SSH into the instance and install Docker:
     [] sudo apt-get update -y sudo apt-get install -y docker sudo service docker start

2.5 AWS CI/CD pipeline

1. **Create CodeBuild project**:
   - Navigate to **CodePipeline** in the AWS console
   - Under the **CodePipeline** console, click on **Build - CodeBuild**, go to the **CodeBuild** console and click by selecting **Create Project**

- Give a Project name (for e.g- **mycodebuildproject**), then under **Source** section, select **GitHub**, there will be 2 options to authenticate **CodeBuild** access to your **GitHub** repository - **OAuth** and **Personal Access Token**, select **Oauth** option then click on **Connect to GitHub**, a window will popup to log in to your GitHub repo.
- Review the permissions, click on **Authorize CodeBuild**, then login into **GitHub** by supplying your credentials, and click **Confirm**
- Select the **Build** environment configuration as **On-demand**, then **Managed Image**, then **EC2**, OS as **Ubuntu - Standard**, **Image** as **standard:5.0**, **version** as **Always use latest image for this runtime version**.
- Move on to **Buildspec** and choose the **Use a buildspec file** option, this will look for the buildspec.yml file in the the GitHub repo which you have already copied
- For the remaining part, leave everything to the default settings and click on **Create Project**

2. **Configure CodePipeline:**

   - Navigate to creating a **New pipeline**, give it a name, proceed to the **Build** stage and select the **CodeBuild** project created in the previous step, create the IAM role required for it and attach IAM role permissions linked to the service when prompted to do, skip the **Deploy** stage since as it won't be required since we only plan to perform scans and not deploy the scanned container images to **ECS** or **EKS** and later click on **Create**
   - Once created, trigger the pipeline build process by selecting the **Trigger pipeline** or **Release change** option in the **CodePipeline** console.
   - After **CodePipeline** has been triggered, it will fetch for the **buildspec.yml** source file by calling it from the GitHub repository located in the main branch which contains instructions segmented in various phases like pre-build, build and post-build, call the Trivy and Grype script files by executing them using **CodeBuild** phase, start the scanning process and output the scan results to the configured **S3** bucket.

2.6 AWS Inspector

1. **Enable Inspector**:

   - Navigate to **Inspector** in the AWS Console.
   - Configure your scanning targets (ECR images or EC2 instances).

2. **Schedule Scans**:

   - Set up a schedule for recurring scans or initiate manual scans from the dashboard.

2.6 AWS CloudWatch

1. **Set Up Monitoring**:
   - Navigate to **CloudWatch** > **Logs** > **Create Log Group**.
   - Attach CloudWatch logs to your EC2 instance.
   - Configure alarms for critical metrics (e.g., resource usage, vulnerabilities).

2.0.1 3. IAM Roles and Policies

- **ECR Access Role**:
  - Attach a policy allowing access to ECR:
    [] { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["ecr:GetDownloadUrlForLayer", "ecr:BatchGetImage", "ecr:BatchCheckLayerAvailability"], "Resource": "*" }] }
- **Inspector Role**:
  - Create a role with `AmazonInspectorFullAccess` attached.
- **S3 Access Role**:
  - Use the `AmazonS3FullAccess` managed policy for the instance or service accessing S3.
- **CloudWatch Role**:
  - Use `CloudWatchAgentServerPolicy` to enable logging from EC2 or ECS.

2.0.2 4. Network Configuration

- **Virtual Private Cloud (VPC)**:
  - Create a VPC with public and private subnets.
  - Ensure proper routing and security groups are configured.
- **Security Groups**:
  - Allow necessary inbound and outbound rules for EC2 communication.

2.0.3 5. Scanning Schedule

- Bi-weekly scans will be performed for each Docker image over a three-month period.
- Use AWS Inspector to automate recurring scans and log results in S3.

2.0.4 6. Resource Specifications

Resource	Type	Configuration
EC2 Instances	t2.medium	20GB Storage, SSH
S3 Bucket	Standard	Secure permissions
ECR Repos	Standard	Scan on Push Enabled

References

DockerHub. Research images. *Accessed: Dec 12, 2024*. Available at: `https://hub.docker.com/r/dh157/research_images_sept_2022`.

DockerHub. Research images. *Accessed: Dec 12, 2024*. Available at: `https://hub.docker.com/r/dh157/research_images_dec_2022`.

Cyberithub. Configuring Trivy on Ubuntu 22.04 Server. *Accessed: Dec 12, 2024*. Available at: `https://www.cyberithub.com/how-to-install-trivy-vulnerability-scanner-on-ubunt`

Lindevs. Installing Grype on Ubuntu. *Accessed: Dec 12, 2024*. Avaialable at: `https://lindevs.com/install-grype-on-ubuntu/`