National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Clint Fernandes
Student ID: 23348089

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Clint Fernandes
Student ID:	23348089
Programme:	Cloud Computing **Year:** 2024 - 2025
Module:	MSc Research Project
Lecturer:	Sean Heeney
Submission Due Date:	11/08/2025
Project Title:	Performance Enhancement and Security Integration in a Multi-Objective Optimization Framework for Docker Image Slimming
Word Count:	**1032** **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: Clint Fernandes

Date: 11/08/2025

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

Forename Surname
Student ID:

1 Section 1

Your first section. Change the header and label to something appropriate.

2 Section 2

Your second section. Change the header and label to something appropriate.

3 Section 3

Your third section. Change the header and label to something appropriate.

References

References should be formatted using APA or Harvard style as detailed in NCI Library Referencing Guide available at https://libguides.ncirl.ie/referencing
You can use a reference management system such as Zotero or Mendeley to cite in MS Word.

Beloglazov, A. and Buyya, R. (2015). Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds, *Concurrency and Computation: Practice and Experience* 27(5): 1310–1333.

Feng, G. and Buyya, R. (2016). Maximum revenue-oriented resource allocation in cloud, *IJGUC* 7(1): 12–21.

Gomes, D. G., Calheiros, R. N. and Tolosana-Calasanz, R. (2015). Introduction to the special issue on cloud computing: Recent developments and challenging issues, *Computers & Electrical Engineering* 42: 31–32.

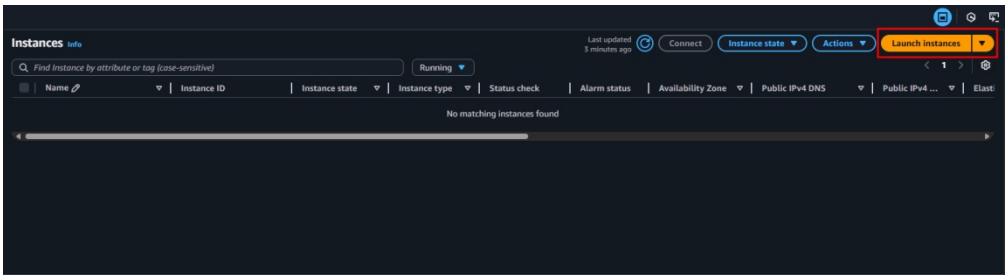
Kune, R., Konugurthi, P., Agarwal, A., Rao, C. R. and Buyya, R. (2016). The anatomy of big data computing, *Softw., Pract. Exper.* 46(1): 79–105.

Setting Up SCALPEL2 Cloud Environment

To submit an image to SCAPEL2 for slimming, we need a local system or environment to act like a user of the tool. This user will create a Docker image out of the 20 NPM packages in the dataset for pre-processing. For the experiment, an EC2 (Elastic Cloud Compute) instance was created with the name SCALPEL2_workspace. The EC2 instance represented a local system from which the Docker image will be pushed to ECR (Elastic Container Registry).

1. Created an EC2 instance named SCALPEL2_workspace:

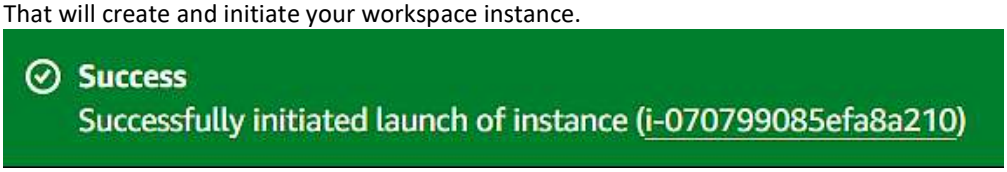
a) Headed over to AWS EC2 dashboard and clicked on 'Launch Instances'.



b) For the details and options to choose from, refer to the table below:

Detail/Option	Values
Name	SCALPEL2_workspace
Amazon Machine Image (AMI)	Amazon Linux 2023 kernel-6.1 AMI
AMI ID	ami-0b6acaa45fec15278
Architecture	64-bit (x86)
Instance type	t3.micro (2 vCPU, 1 GiB Memory)
Key pair	Generated and downloaded
Network settings	Left the default options
Storage	20Gib, gp3 type
Advanced details	Left the default options

c) That will create and initiate your workspace instance.

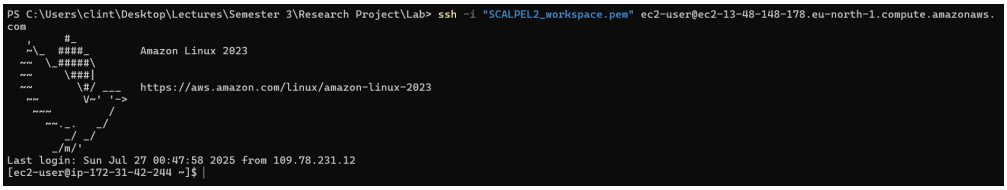


d) Changed the permission of the private key file to ensure its not publicly available.

```
chmod 400 "SCALPEL2_workspace.pem"
```

e) Connected with the instance using the SSH command.

```
ssh -i "SCALPEL2_workspace.pem" ec2-user@ec2-16-171-149-132.eu-north-1.compute.amazonaws.com
```



2. Install Docker:

Documentation: [Installing Docker to use with the AWS SAM CLI]

a) Updated the installed packages on the workspace instance

```
sudo yum update -y
```

b) Installed the most recent Docker Community Edition package

```
sudo yum install -y docker
```

c) Started and enabled the Docker service

```
sudo systemctl start docker
sudo systemctl enable docker
```

d) Added the ec2-user to the docker group to run Docker commands without using sudo

```
sudo usermod -aG docker ec2-user
```

e) Verified that the ec2-user can run Docker commands without using sudo.

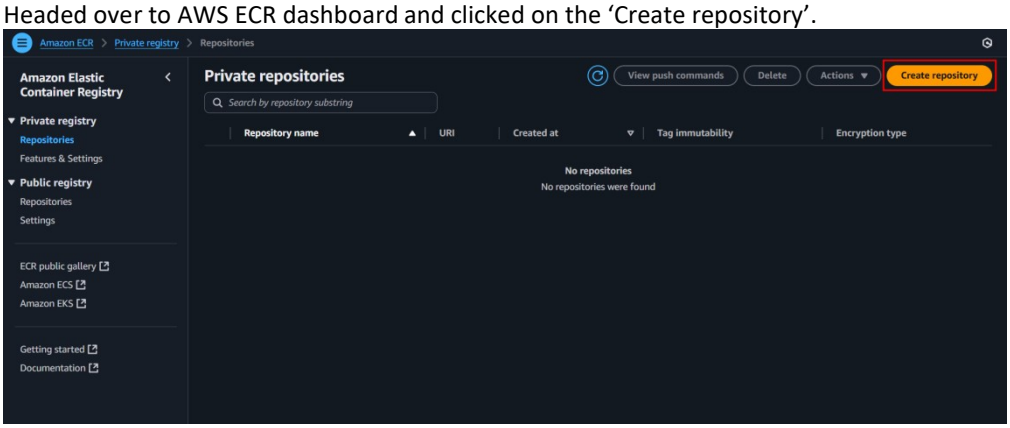
```
docker ps
```

Displayed the below output

```
[ec2-user@ip-172-31-42-244 ~]$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

3. Created a Private ECR Registry:

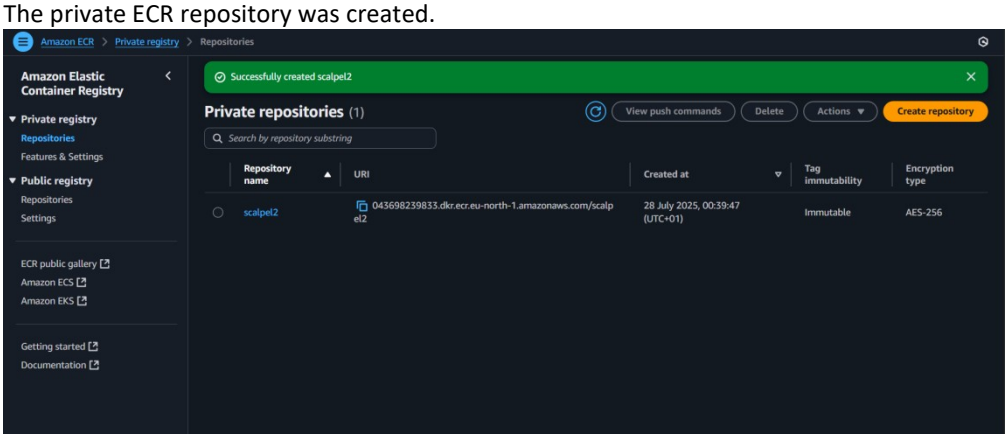
a) Headed over to AWS ECR dashboard and clicked on the ‘Create repository’.



b) For the details and options to choose from, refer to the table below:

Detail/Option	Values
Repository name	043698239833.dkr.ecr.eu-north-1.amazonaws.com/**scalpel2**
Image tag mutability	Immutable
Encryption settings	Left the default options
Image scanning settings	Left the default options

c) The private ECR repository was created.



d) The repository URI is:

```
043698239833.dkr.ecr.eu-north-1.amazonaws.com/scalpel2
```

4. Built Docker image from dataset package:

a) SSH into the SCALPEL2_workspace

```
ssh -i "SCALPEL2_workspace.pem" ec2-user@ec2-16-171-149-132.eu-north-1.compute.amazonaws.com
```

b) Enter the **datasets** directory

```
cd datasets/
```

c) Choose to build the **deepmerge** package Docker image **with entry point** and with **node-current-slim** image as a base image. Accordingly entered the right directory.

```
cd with\ entry\ point/node-current-slim/deepmerge/
```

d) Use the **ls** command to check if the respective Dockerfile exists

```
ls -l
```

Gave the below output:

```
[ec2-user@ip-172-31-42-244 deepmerge]$ ls
Dockerfile  target-project
```

e) Ran the Docker build command to build the image.

```
docker build -t scalpel:deepmerge_wep_ncs_original_image .
```

The abbreviations **wep** stand for **with entry point**, **woep** stands for **without entry point**, **nc** stands for **node current**, and **ncs** stand for **node current slim**.

Gave the below output:

```
[ec2-user@ip-172-31-42-244 deepmerge]$ docker build -t scalpel:deepmerge_wep_ncs_original_image .
[+] Building 13.3s (10/10) FINISHED                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                           0.0s
 => => transferring dockerfile: 810B                                                                           0.0s
 => [internal] load metadata for docker.io/library/node:current-slim                                           1.4s
 => [internal] load .dockerignore                                                                              0.0s
 => => transferring context: 2B                                                                                0.0s
 => [1/5] FROM docker.io/library/node:current-slim@sha256:36ae19f59c91f3303c7a648f07493fe14c4bd91320ac8d898416  4.2s
 => => resolve docker.io/library/node:current-slim@sha256:36ae19f59c91f3303c7a648f07493fe14c4bd91320ac8d898416  0.0s
 => => sha256:36ae19f59c91f3303c7a648f07493fe14c4bd91320ac8d898416327bacf1bbfa 5.20kB / 5.20kB                 0.0s
 => => sha256:59dc16b4f436d8b03119f93de0d74236cba204ebe474d6545dfb4a9b0b1c2676 1.93kB / 1.93kB                 0.0s
 => => sha256:6c50ff79f34e4331d02817efffbe76aaf496c29d16db691ba4a3bf0c7ef1b7fb 6.79kB / 6.79kB                 0.0s
 => => sha256:59e22667830bf04fb35e15ed9c70023e9d121719bb87f0db7f3159ee7c7e0b8d 28.23MB / 28.23MB               0.5s
 => => sha256:08e49ae6d850aafe12ddf2411a83e99c1b224eaafba05aa7e0dee762b6b28d7c 3.31kB / 3.31kB                 0.5s
 => => sha256:6d8e03ec47c613c5f286f559ec47f99a10eb4d7f348c6da120b5604ac179955e 51.12MB / 51.12MB               1.0s
 => => sha256:45c9f1fffd4e88230ec2aa578d11797df2222d19fdbb8309b1c2ab3fbd16288c 1.71MB / 1.71MB                 0.8s
 => => extracting sha256:59e22667830bf04fb35e15ed9c70023e9d121719bb87f0db7f3159ee7c7e0b8d                      1.3s
 => => sha256:02c93d07029f26c7175b57998c8622d63f3645d5da4f31f76d2f7ca296e6ff6b 448B / 448B                     0.8s
 => => extracting sha256:08e49ae6d850aafe12ddf2411a83e99c1b224eaafba05aa7e0dee762b6b28d7c                      0.0s
 => => extracting sha256:6d8e03ec47c613c5f286f559ec47f99a10eb4d7f348c6da120b5604ac179955e                      2.0s
 => => extracting sha256:45c9f1fffd4e88230ec2aa578d11797df2222d19fdbb8309b1c2ab3fbd16288c                      0.1s
 => => extracting sha256:02c93d07029f26c7175b57998c8622d63f3645d5da4f31f76d2f7ca296e6ff6b                      0.0s
 => [internal] load build context                                                                              0.0s
 => => transferring context: 102.83kB                                                                          0.0s
 => [2/5] RUN npm config set registry https://registry.npmmirror.com     && mkdir -p /opt/target-project     &  1.1s
 => [3/5] COPY ./target-project /opt/target-project                                                            0.1s
 => [4/5] WORKDIR /opt/target-project                                                                          0.0s
 => [5/5] RUN npm config set prefix "/opt/npm_pkgs/node_global"     && npm config set cache "/opt/npm_pkgs/nod  5.8s
 => exporting to image                                                                                         0.6s
 => => exporting layers                                                                                        0.5s
 => => writing image sha256:41a5135609b77aabe6953071bf41532c50ffc0f55d76d28a78512f993b513d30                   0.0s
 => => naming to docker.io/library/scalpel:deepmerge_wep_ncs_original_image                                    0.0s
[ec2-user@ip-172-31-42-244 deepmerge]$
```

5. Pushed a Docker image to AWS ECR:

Documentation: [Pushing a Docker image to an Amazon ECR private repository]

a) Configured the AWS CLI with the right credentials. The access key and secret access key can be found in the access key sections under AWS IAM credentials of the user.

```
aws configure
```

Gave the below output:

```
[ec2-user@ip-172-31-42-244 deepmerge]$ aws configure
AWS Access Key ID [None]: AKIAQULE6HFM6ZAHCGOX
AWS Secret Access Key [None]: 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
Default region name [None]: eu-north-1
Default output format [None]:
[ec2-user@ip-172-31-42-244 deepmerge]$
```

b) Retrieved an authentication token and authenticated the Docker client to the ECR

aws ecr get-login-password --region eu-north-1 | docker login --username AWS --password-stdin 043698239833.dkr.ecr.eu-north-1.amazonaws.com

Gave the below output:

```
[ec2-user@ip-172-31-42-244 deepmerge]$ aws ecr get-login-password --region eu-north-1 | docker log
in --username AWS --password-stdin 043698239833.dkr.ecr.eu-north-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-42-244 deepmerge]$
```

c) Tagged the local Docker image with the ECR repository URI

docker tag scalpel:deepmerge_wep_ncs_original_image 043698239833.dkr.ecr.eu-north-1.amazonaws.com/scalpel2:deepmerge_wep_ncs_original_image

d) Pushed newly tagged image to the ECR

docker push 043698239833.dkr.ecr.eu-north-1.amazonaws.com/scalpel2:deepmerge_wep_ncs_original_image

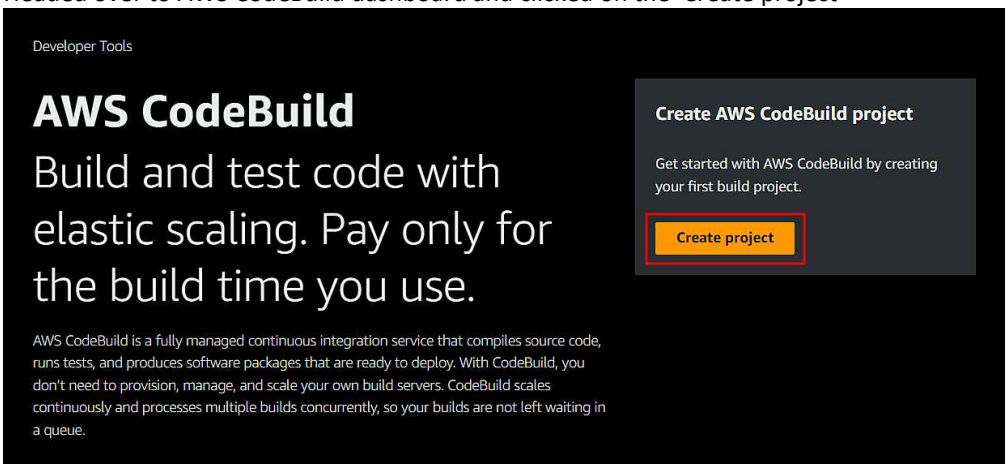
Gave the below output:

```
[ec2-user@ip-172-31-42-244 deepmerge]$ docker push 043698239833.dkr.ecr.eu-north-1.amazonaws.com/s
calpel2:deepmerge_wep_ncs_original_image
The push refers to repository [043698239833.dkr.ecr.eu-north-1.amazonaws.com/scalpel2]
1ba26b236153: Pushed
5f70bf18a086: Pushed
52fc36b9a20b: Pushed
065a156986d7: Pushed
5a4f5472da49: Pushed
53a0f4bc03e4: Pushed
c2959f3fcaa3: Pushed
c3cdd708ab79: Pushed
7cc7fe68eff6: Pushed
deepmerge_wep_ncs_original_image: digest: sha256:7b290bfdec2c0329ceb62eef58fa33d1831b5838be0dc7a6e
1f05f1768fb24a3 size: 2204
[ec2-user@ip-172-31-42-244 deepmerge]$
```

e) Verified if pushed image was visible in the ECR repository images

6. Created a CodeBuild project to run SCALPEL1:

a) Headed over to AWS CodeBuild dashboard and clicked on the 'Create project'



b) For the 'Project configuration' steps details and options, refer to the table below:

Detail/Option	Values
Project name	codebuild_scalpel1_project
Project type	Default project
Source 1 - Primary	GitHub

c) Clicked on 'Manage account credentials'. It opened another tab.

d) For the 'Manage default source credential' steps details and options, refer to the table below:

Detail/Option	Values
Credential type	Personal access token
Service	CodeBuild
GitHub personal access token	**********

e) Enter the details and click 'Save'. It closed the new tab and returned back to the new CodeBuild project tab.

f) For the 'Source 1 - Primary' steps details and options, refer to the table below:

Detail/Option	Values
Repository	Repository in my GitHub account
Search for	https://github.com/clintfernandes21/scalpel2.git
Source version - optional	running_scalpel1

g) For the 'Primary source webhook events' steps details and options, refer to the table below:

Detail/Option	Values
Rebuild every time a code change is pushed to this repository	Unchecked

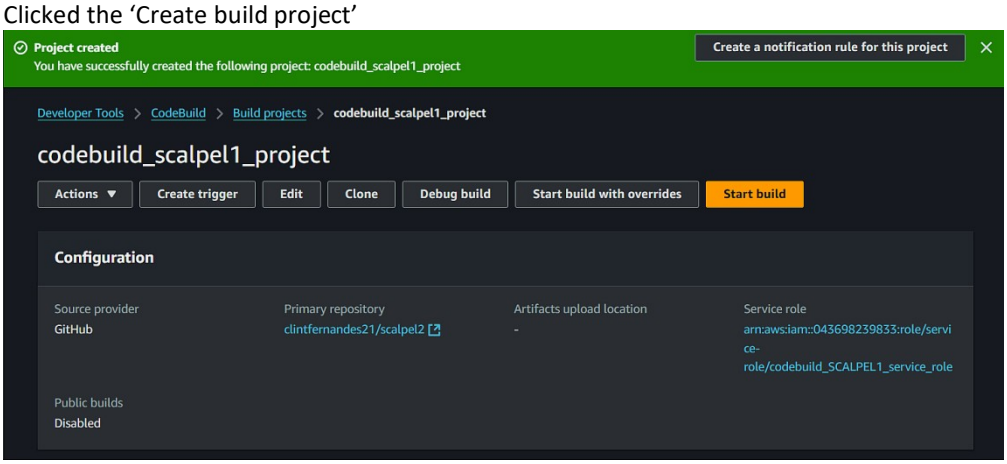
h) For the 'Environment' steps details and options, refer to the table below:

Detail/Option	Values
Service role	New service role
Role name	codebuild_SCALPEL1_service_role
Enable this flag if you want to build Docker images or want your builds to get elevated privileges	Checked
Compute	2 vCPUs, 4 GiB memory

i) For the 'Buildspec' steps details and options, refer to the table below:

Detail/Option	Values
Build specifications	Use a buildspec file

j) Clicked the 'Create build project'



k) Created service instance details:

Detail	Values
Service role	https://eu-north-1.console.aws.amazon.com/go/view?arn=arn%3Aaws%3Aiam%3A%3A043698239833%3Arole%2Fservice-role%2Fcodebuild_SCALPEL1_service_role&source=codesuite
Project ARN	arn:aws:codebuild:eu-north-1:043698239833:project/codebuild_scalpel1_project
Repository	clintfernandes21/scalpel2
Source version (branch)	running_scalpel1

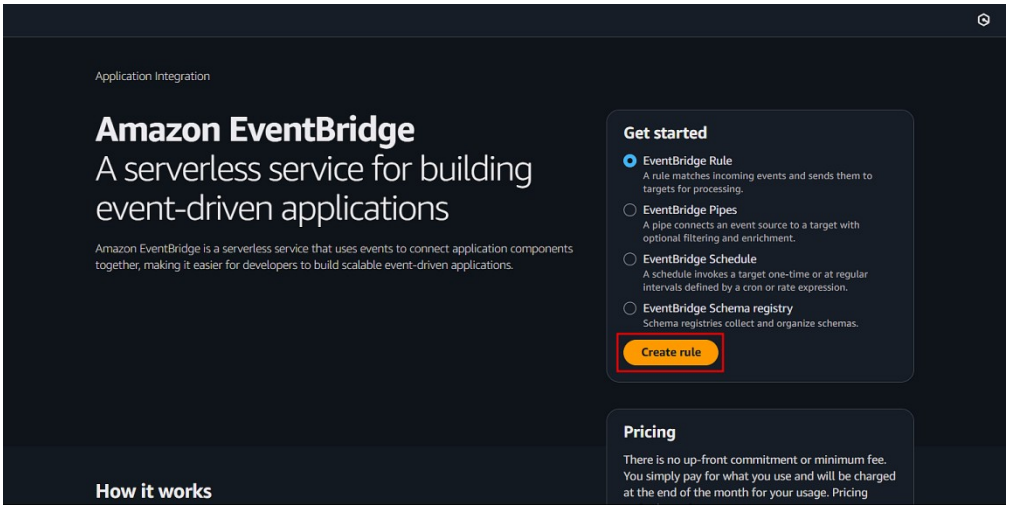
l) Updated the CodeBuild service role permissions to allow it to access AWE ECR. Add the below lines to the existing permissions at the end.

```
{
   "Effect": "Allow",
   "Action": [
    "ecr:GetAuthorizationToken",
    "ecr:BatchGetImage",
    "ecr:GetDownloadUrlForLayer"
   ],
   "Resource": "*"
}
```

7. Created an EventBridge Rule:

AWS EventBridge can detect an image push to ECR, via a rule that listens for a **PutImage** event. When the rule is triggered, it can invoke the AWS CodeBuild service, which will run the SCALPEL2 tool with the pushed image.

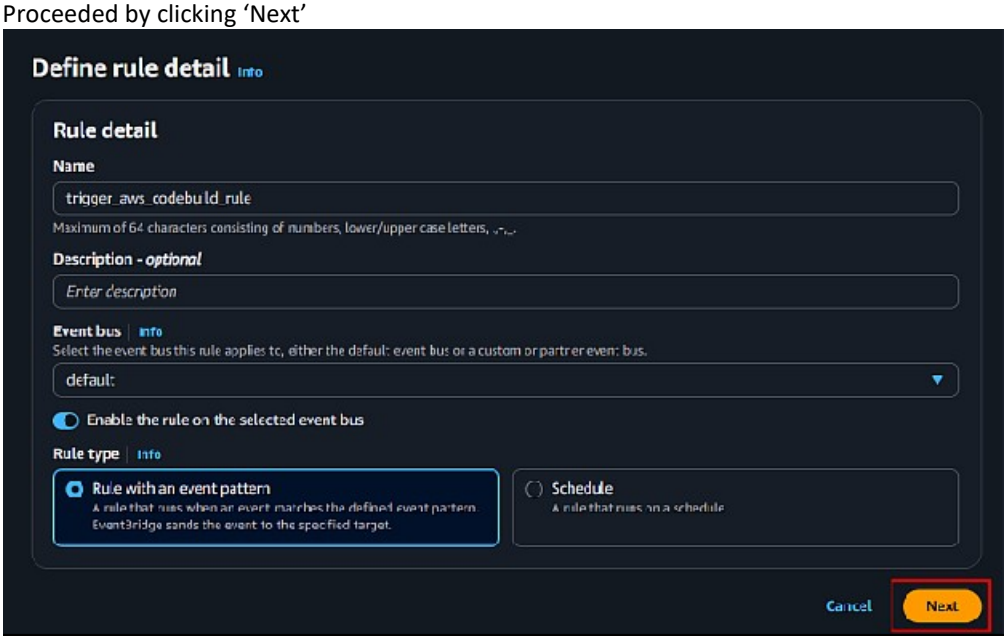
a) Headed over to AWS EventBridge dashboard and clicked on the 'Create rule'



b) For the 'Define rule details' steps details and options, refer to the table below:

Detail/Option	Values
Name	trigger_codebuild_projects_rule
Event bus	default
Enable the rule on the selected event bus	enabled
Rule type	Rule with an event pattern

c) Proceeded by clicking 'Next'



d) For the 'Build event pattern' steps details and options, refer to the table below:

Detail/Option	Values
Event -> Event source	AWS events or EventBridge partner events
Event pattern -> Creation method	Custom pattern (JSON editor)

e) For the event pattern, refer to the JSON snippet below:

```
{
  "source": ["aws.ecr"],
  "detail-type": ["ECR Image Action"],
  "detail": {
    "action-type": ["PUSH"],
    "result": ["SUCCESS"],
    "repository-name": ["scalpel2"]
  }
}
```

f) Proceeded by clicking 'Next'

g) For the 'Select target(s)' steps details and options, refer to the table below:

Detail/Option	Values
Target1 -> Target types	AWS service
Target1 -> Select a target	Codebuild project
Target1 -> Project ARN	arn:aws:codebuild:eu-north-1:043698239833:project/codebuild_scalpel1_project
Target1 -> Execution role	Create a new role for this specific resource
Target1 -> Role name	Amazon_EventBridge_Invoke_CodeBuild_158242784
Target1 -> Configure target input	Input transformer

h) Clicked on 'Configure Input transformer'. A dialog box opened

i) For the 'Input path' enter

```
{"tag":"$.detail.image-tag"}
```

j) For the 'Template' enter

```
{
  "environmentVariablesOverride": [
    {
      "name": "IMAGE_TAG",
      "value": "<tag>",
      "type": "PLAINTEXT"
    }
  ]
}
```

k) Then clicked the 'Confirm' button.

l) Proceeded without any more changes to the page. Clicked on 'Skipped to Review and create'.

m) Reviewed the details and clicked ‘Create rule’