

Configuration Manual

MSc Research Project In Cloud Computing

Ravina Mestry Student ID: x22177264

School of Computing National College of Ireland

Supervisor:

Vikas Sahni

National College of Ireland



MSc Project Submission Sheet

School of Computing

Student Name:	Ravina Mestry
Student ID:	x22177264
Programme:	MSc in Cloud Computing Year:2023
Module:	Research Project in Cloud Computing
Lecturer:	Vikas Sahni
Due Date:	14/08/2023
Project Title:	Securing the Speed: Balancing Security and Deployment Velocity in DevOps

Word Count:2165....... Page Count:16.......

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.

<u>ALL</u> 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:	Ravina Mestry
Date:	10/08/2023

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

Ravina Mestry Student ID: x22177264

1 Introduction

The system requirements, setup, software, and installation specifications used in this research will be better understood by readers of this configuration manual. Additionally, a detailed explanation of the procedures to be followed when carrying out this research project is provided in this manual. Table 1 lists the tools version and URLs discussed in detail in Section 3.

Prerequisites

- AWS and Django Python knowledge.
- AWS, Docker, and GitHub login.
- Infrastructure as code knowledge.

No.	Tools	Version	URL
1	Terraform	1.5.3	Overview x22177264-research-project-webapp nci-research- project Terraform Cloud
2	Docker	3.8	ravinamestry/x22177264_ravina_mestry general Docker Hub
3	OWASP ZAP	2.13.0	http:// <x22177264_ ravina_mestry_research_project_zap_<br="">PUBLIC_IP>:8080/</x22177264_>
4	Snyk	-	https://docs.snyk.io/integrations/ci-cd-integrations/github- actions-integration
5	Datadog	v7	Agent 7 Datadog (datadoghq.eu)
6	SonarQube	3	http:// <x22177264_ ravina_mestry_research_project_sonarqube<br="">_PUBLIC_IP>:9000/</x22177264_>
7	GitHub	2.34.1	ravinamestry/x22177264_ravina_mestry_research_project (github.com)

Table 1: AWS EC2 Instance Specification

2 Before you begin

2.1 Web Application

The web application Employee Contacts was developed in Django Python using the Django framework. Django offers the first setup for a virtual environment. In addition to CRUD features like create, update, and delete shown in Figure 1, Employee Contacts also offer non-CRUD functionality like data validation for fields in the add new employee page form. The database that is utilized by the application is SQLite. The employee contacts web app is

installed on Gunicorn wsgi application server and SQLite database is migrated using below commands.

python3 manage.py makemigrations python3 manage.py migrate

Employee Ad	dition Successful!					×
		Er	nployees -	List		
FirstName	LastName	Email	Mobile	Department	Role	Action
Ravina	Mestry	ravina@employee.com	353861112222	Tech	Project Manager	Update Delete
John	William	John.william@gmail.com	353861785962	IT	Support	Update Delete
Marc	Evans	Marc.evans@icloud.com	353898967415	IT	HR	Update Delete
Arthur	Ryan	arthur.ryan@gmail.com	353874569685	Purchasing	Intern	Update Delete
Damon	Trevino	damon.Trevino@icloud.com	353864875962	Developer	Junior Developer	Update Delete
Luke	Smith	luke.smith@gmail.com	353965874568	Testing	Tester	Update Delete

Figure 1: Web Application

- Created a virtual Python environment on local machine to install Django.
- Activated the virtual environment.

cd /mnt/c/work/workspace/django-workspace python3 -m venv ravina_mestry_research_project_webapp _gunicorn _venv source ravina_mestry_research_project_webapp_gunicorn_venv/bin/activate

- Installed Python 3.10.6.
- Used pip to install Django 3.2.13.
- Started the Django project and started with below command.
- cd ravina_mestry_research_project_webapp

python3 manage.py runserver 8080

- After adding the URL generated by running the server in settings.py 'ALLOWED_HOSTS' the Django application is shown running successfully.
- The web application is then deployed on AWS EC2 instance in Development (Dev), Staging (Stage) and Production (Prod) using Docker image.

3 Tools/Cloud-based services

3.1 AWS EC2

- AWS EC2 instances are used for deploying dev, stage and prod environment for running the Web application, also for installing SonarQube and Zap discussed in Section 3.4 and 3.6 respectively.
- AWS Instances are created using Terraform, discussed in Section 3.2.

Inst	ances (5) Info					Conn	ect Ins	tance state 🔻	Actions 1	Launch insta	inces
Q	Find instance by attribute or tag (case-sensitive)										
<u>x22</u>	177264 X Clear filters										1
	Name 🗸 🗸	Instance ID	Instance state		Instance type 🔻	Status check	Alarm statu	s Availabil	ity Zone 🔻 🛛	Public IPv4 DNS	~
	x22177264_ravina_mestry_research_project_zap	i-Oeee18a01ed1d8cd5		ଭ୍ର	t2.medium		No alarms	+ us-east-1	e	ec2-54-234-97-207.	.co
	x22177264_ravina_mestry_research_project_prod	i-01c4fc82209a555c7		ଭ୍ର	t2.micro		No alarms	+ us-east-1	e	ec2-18-209-180-198	8.co
	x22177264_ravina_mestry_research_project_dev	i-085c3ce7cdc8a801a		ଭ୍ର	t2.medium		No alarms	+ us-east-1	e	ec2-34-239-250-35.	.co
	x22177264_ravina_mestry_research_project_stage	i-08f2ac23d2c623130		ଭ୍ର	t2.micro		No alarms	+ us-east-1	e	ec2-100-27-12-158.	.co
	x22177264_ravina_mestry_research_project_sonarqube	i-Oddf8b3af366ae6a4		ଭ୍ର	t2.medium		No alarms	+ us-east-1	e	ec2-54-157-183-210	0.co

Figure 2: AWS EC2 Instances

• The Terraform-generated EC2 instances for this project are listed shown in Figure 2 and details of Instance Type and AMI ID are listed in Table 2.

AWS Instance Name	Instance	AMI ID
	Туре	
x22177264_ravina_mestry_research_project_sonarqub	t2.mediu	ami-
e	m	04505e74c0741db8
		d
x22177264_ravina_mestry_research_project_zap	t2.mediu	ami-
	m	04505e74c0741db8
		d
x22177264_ravina_mestry_research_project_dev	t2.micro	ami-
		053b0d53c279acc90
x22177264_ravina_mestry_research_project_stage	t2.micro	ami-
		053b0d53c279acc90
x22177264_ravina_mestry_research_project_prod	t2.micro	ami-
		053b0d53c279acc90

Table 2: AWS EC2 Instance	Specification
---------------------------	---------------

3.2 Terraform

- new Created Terraform • nci-research-Organization project and workspaces x22177264-research-project, x22177264-research-projectsonarqube and x22177264research-project-zap shown in Figure 3.
- Saved the AWS variables for above three workspaces in Terraform shown in Figure 4.



Figure 3: Terraform organization and Workspaces

- In main.tf for Terraform mentioned the EC2 instance details in Figure 5 for creating development, Staging, and production environments with the Key name, Security group ID and Tag name.
- Terraform also notes Public_IP for the EC2 instances and terraform_create.yml saving the Public_IP in GitHub Actions in Figure 6.
- The Public_ID noted in Secrets of GitHub in Figure 7 is retrieved by deploy-dev.yml and deploy.yml to deploy the Web Application to respective environments using Docker.
- Docker Images which consist of Web Application are copied to the Public_IP saved in GitHub Secrets.

Workspace variables (4)

Variables defined within a workspace a

Key

AWS_ACCESS_KEY_ID

AWS_SECRET_ACCESS_KEY SENSITIVE

AWS_SESSION_TOKEN SENSITIVE

Figure 4: Terraform AWS Variables

#dev instance creation	
<pre>resource "aws_instance" "x22177264_ravina_mestry_research_project_dev" {</pre>	
amī = "amī-04505e74c0741db8d"	
#ami = "ami-053b0d53c279acc90"	
<pre>instance_type = "t2.medium"</pre>	
<pre>subnet_id = "subnet-07f96f862eb5dbc50"</pre>	
associate_public_ip_address = true	
<pre>key_name = "x22177264_ravina_mestry_research_project_key"</pre>	
<pre>vpc_security_group_ids = ["sg-00f05bacad60c315d"]</pre>	
tags = {	
<pre>Name = "x22177264_ravina_mestry_research_project_dev"</pre>	
}	



<pre>output "x22177264_ravina_mestry_research_project_de value = "\${aws_instance.x22177264_ravina_mestry_r }</pre>	ev_public_ip" { research_project_dev.pub	olic_ip}"
Resources 3 Outputs 3		Curre
NAME ↓	ТҮРЕ	VALUE
x22177264_ravina_mestry_research_project_dev_public_ip	string	"52.91.148.130" 🛃
x22177264_ravina_mestry_research_project_prod_public_ip	string	"18.209.180.198"
x22177264_ravina_mestry_research_project_stage_public_ip	string	"100.27.12.158" 民

Figure 6: Terraform saving Public_IP in GitHub Secrets

Crea succe	a tin ede	d last week in 2m 39s
~ (ø	Terraform Apply
	52	aws_instance.x22177264_ravina_mestry_research_project_dev: Still modifying [1m20s elapsed]
		aws_instance.x22177264_ravina_mestry_research_project_dev: Still modifying [1m30s elapsed]
	54	aws_instance.x22177264_ravina_mestry_research_project_dev: Still modifying [1m40s elapsed]
		aws_instance.x22177264_ravina_mestry_research_project_dev: Modifications complete after 1m41s [id=i-085c3ce7cdc8a801a]
	57	Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
		outputs:
		x22177264_ravina_mestry_research_project_stage_public_ip = "100.27.12.158"
		x22177264_ravina_mestry_research_project_dev_public_ip = "52.91.148.130"
	52	<pre>x22177264_ravina_mestry_research_project_prod_public_ip = "18.209.180.198"</pre>

Figure 7: Terraform creating Instances for Dev/Stage/Prod

3.3 Docker

- Created the repository ravina-mestry/x22177264_ravina_mestry to save the Docker Image for Web app shown in Figure 9.
- Docker Image is built and pushed to Docker Hub by Terraform in .GitHub/workflows/build.yml workflow in Figure 8.



Figure 8: Docker Image Tag Specification

	Jocker hub Q Search Docker Hub	Explore Rep	ositories Organizations Hel	p 👻	Upgrade 🥡 ravinamestry 🗸
ravin	amestry Repositories x22177264	I_ravina_mestry Tags			Using 0 of 1 private repositories. Get more
Gene	ral Tags Builds Collaborator	s Webhooks Settings			
	Sort by Newest - Filter	Tags Q	Go to Advanced Image Management	Delete	
	TAG <u>ncirl_cloud_research_webapp_x2217</u> Last pushed 2 days ago by <u>ravinamestry</u>	7264_ravina_mestry			docker pull ravinamestry/x221772
	DIGEST <u>8d1c99f3e791</u>	OS/ARCH linux/amd64	SCANNED	LAST PULL 20 hours ago	COMPRESSED SIZE [©] 69 MB

Figure 9: Docker Image pushed in Docker Hub

• Docker-compose consists of Docker Image to be pulled and ran in Docker container which runs the Web app on Dev/Stage/Prod environments in Figure 10 using below command. Docker login is stored in GitHub Action Secrets.

sudor docker-compose up -d



Figure 10: Docker Image ran on AWS Instance for Dev/Stage/Prod

3.4 SonarQube

- Created AWS EC2 t2.medium Instance for SonarQube using Terraform in Section 3.1.
- Added the configuration into docker-compose.yaml.
- Increased the Elasticsearch limit by adding the following command in installdocker.sh.

sudo sysctl -w vm.max_map_count=262144

• Built and started the docker containers.

sudo docker-compose up -d

- Navigated to <InstanceIP>:9000 and created the project x22177264_ravina_mestry and configured for GitHub Actions by creating SONAR_TOKEN and SONAR_HOST_URL secrets and sonar-project.properties file in GitHub repository.
- Ran the .github/workflows/build_sonarqube.yml to show the analysis of code in Figure 11.

sonarQube Projects Issues Rules Quality Profiles Quality Gates	Administration More Q				
☆ x22177284_ravina_mestry / 🚦 main 💌 🕥 🕷				Versi	
Overview Issues Security Hotspots Measures Code Activity			F	Project Settings ~	
Quality Gate Status T	Measures New Code Overall Code				
Passeu	in Reliability 3 Bugs	D	Maintainability 32 Code Smells	۸	
	Security Vulnerabilities	۸	Security Review 15 Security Hotspots *	E	Figure 11.
e-b-3 You Housed even core	Coverage 0.0% Coverage Coverage on 12k Lines to cover – Unit Tests	0	Duplications 1.1% Duplications Duplications on 5.9k Lines 2 Duplicated Blocks	۲	SonarQube Code analysis

3.5 GitHub and GitHub Action

- Created GitHub repository x22177264_ravina_mestry _research_project.
- Added the code for webapp discussed in Section 2.1.
- Added terraform/main.tf to create infrastructure in AWS EC2 in Section 3.2.
- Included SonarQube scan, Super-Linter, Snyk scan, and Datadog monitoring tools on the CI-CD pipeline.
- Created Workflows shown in Figure 12 for creating and destroying infrastructure, building Docker Image, and deploying the web app in Dev/Stage/Prod environments.

<pre>x221//264_ravina_mestry_research_project / .github / workflows /</pre>
🕕 ravinamestry Update build_sonarqube.yml 🗸
🖿
🗅 build.yml
build_sonarqube.yml
🗅 deploy-dev.yml
C deploy-stage.yml
🗅 deploy.yml
C terraform_create.yml
terraform_create_sonarqube.yml
L terraform_create_zap.yml
C terraform_destroy.yml
terraform_destroy_sonarqube.yml
terraform_destroy_zap.yml

Figure 12: GitHub Actions Workflows

3.6 ZAP

- Created the AWS EC2 t2.medium Instance for Zap using Terraform discussed in Section 3.1.
- Added owasp/zap2docker-stable in Docker file with entry point zap-webswing.sh.
- Ran the docker containers with the ports 8080 or 8090 in Docker-compose.yml.

3.6.1 Steps to run a Quick Start Automated scan

- Started ZAP <x22177264_ravina_mestry_research_project_zap_PUBLIC_IP> and clicked the Quick Start tab of the Workspace.
- Clicked the large Automated Scan button.
- In the URL to attack text box, entered the full URL of the web application.
- Clicked the Attack.
- Figure 13 shows the scanner attacking passively the web application.

💟 Untitled Session - 20230718-164132 - OWASP ZAP 2.13.0 🕥 🖃 💷 💌						
File Edit View Analyse Report Tools Impor	Export Online Help					
Standard Mode 🗸 📄 😖 📄 🛍 😭 🕵)) 🖉 💥 🖩				
🚱 Sites 🕂	✓ Quick Start	r 🛨				
Contexts	This screen allows you to launch an automated scan against an application URL below and press 'Attack'.	- just enter its				
Ontexts Ontext	Please he aware that you should only attack applications that you have he	en specifically been				
	given permission to test.	in specifically been				
	LIBL to attack: http://3.94.185.249/	Select				
	Use traditional spider:					
	Use ajax spider: with Firefox Headless 🗸					
	Attack Stop					
	Progress: Actively scanning (attacking) the URLs discov	vered by th				
🛗 History 🔍 Search 💾 Alerts 📄 Outp	t 🕷 Spider 👌 Active Scan 📌 🕱 🕂					
IIII New Scan Progress: 0: http://3.94.185.249	🕐 🔟 🔲 🌃 🍼 Current Scans: 1 Num Requests: 118 New Alerts:	0 🖉 Export 🚳				
		C TUPPET ONO				
Sent Messages Filtered Messages						
ID Req. Timestamp Resp. Timestamp Meth	URL Code Reason RTT Size Resp. He	ea Size Resp. Bo 🛱				
230 //10/23, 0.10.42 //10/23, 0.10.42 FUSI IN	p.//3.34.105.249/update/1/ 200 OK 2.430 bytes	2 958 butos				
257 //10/25, 8:10:42 //18/25, 8:10:42 POSI h	p://3.94.185.249/undsta/1/ 500.internal 2, 259 bits	2,908 bytes				
261 7/18/23 8·10·42 7/18/23 8·10·42 POST h	m//3.94.185.249/create/ 200.0K 2.422 bytes	2 846 bytes				
263 7/18/23, 8:10:42 7/18/23, 8:10:42 POST h	p://3.94.185.249/update/1/ 500 Internal 1 258 bytes	145 bytes				
Alerts 📕 0 🏴 1 🏳 2 🎮 3 Main Proxy: 0.0.0.0	3090 Current Scans 🌞 0 🐼 2 🁌 1 🎯 0	≥ 0 ₩ 0 ₩ 0				

Figure 13: GitHub Actions Workflows

3.7 Snyk

- Created Snyk account with organization name 'ravinamestry' and generated Snyk API Token.
- Integrated Snyk with GitHub Actions and Docker Hub. Enables Repository access for GitHub Actions and enables Detect application vulnerabilities for Docker Hub in Figure 15.
- Created GitHub Action Secret for SNYK_TOKEN which is used in Figure 14 env section in .github/workflows/build.yml workflow for Scanning the web app.

Repository access Whether Snyk can access private repos. Changing this setting affects existing Projects.	Allow access to private repositories
Detect application vulnerabilities	
Snyk can scan applications in your container images from your regist	try and surface vulnerabilities.
Save changes	

Figure 14: Snyk GitHub Actions and Docker Hub Integration

- Snyk scans the image ravinamestry/x22177264_ravina_mestry:ncirl_cloud_research _webapp_x22177264_ravina_mestry for high severity vulnerabilities mentioned in Figure 15.
- Used --sarif-file-output and the GitHub SARIF upload action, when GitHub creates code scanning alerts in a repository using information from Static Analysis Results Interchange Format (SARIF) files, SARIF files are uploaded to a repository using GitHub Actions.
- Continue-on-error is true so that when Snyk Action fails when vulnerabilities are found this would not prevent the SARIF upload action from running.



Figure 15: Snyk configuration

• Figure 16 shows the Docker image and GitHub repo scanned for vulnerabilities and Snyk tracks and flags Pull Requests in the top-most vulnerable projects with severity levels critical, high, medium, and low.

(ii) snyk ≕	ravinamestry > Dashboard						
organization R ravinamestry ↔→ Dashboard	Pending tasks Snyk tracks and flags Pull Requests (PRs) in the top-most vulnerable projects						
☐ Projects ☆ Integrations	PROJECT 🗢		FIXABLE ISSUES 🜩	ACTIONS			
د Members	ravinamestry/x22177264_ravina_mestry_re	search_project	5 C 10 H 13 M 0 L	Fix vulnerabilities			
U Setungs	ravinamestry/x22177264_ravina_mestry_re	search_project	1 C 5 H 3 M 0 L	Fix vulnerabilities			
	Vulnerable projects Projects with vulnerabilities detected						
	PROJECT 🖨	TESTED 🌩	ISSUES 🌩	ACTIONS			
	ravinamestry/x22177264_ravina_mestry_ research_project:webapp/Dockerfile	an hour ago	23 C 49 H 31 M 76 L	Fix vulnerabilities			
	ravinamestry/x22177264_ravina_mestry: ncirt_cloud_research_webapp_x22177264 _ravina_mestry	3 hours ago	23 C 49 H 31 M 76 L				
	ravinamestry/x22177264_ravina_mestry: cirl_cloud_research_webapp_x22177264 _ravina_mestry./usr/src/app/requirement s.txt	4 hours ago	1 C 5 H 3 M 0 L				
	ravinamestry/x22177264_ravina_mestry_ research_project:webapp/requirements.tx	a day ago	1 C 5 H 3 M 0 L	Fix vulnerabilities			

Figure 16: Snyk dashboard

3.8 Datadog

3.8.1 Set up for Tracing on GitHub Actions Workflows

- Configured the GitHub App name 'Datadog ResearchProject' in Datadog integration in Figure 17.
- Edited the Permissions to grant Actions: Read access.
- Configured tracing for GitHub Actions for Enabling CI Visibility for the research repository in Figure 18.
- The Pipelines page in Datadog shows Pipelines and Pipeline Execution with duration and CI status in Figure 19.

Create GitHub App	
GitHub App name	
Datadog - ResearchProject	
Don't worry, you can always change this later.	
Create GitHub App for ravinamestry	

Figure 17: Created GitHub App in Datadog



Figure 18: Enabling CI Visibility in Datadog

URL: <u>Set up Tracing on GitHub Actions Workflows (datadoghq.com)</u>

Velcome, Ravinat Get Started • You have 14 days left in your t									
℃ CI Pipelines ∨ Pipelines	Pipeline Executions	Dashboard 🗹					1w Past 1 Week		*
Q Filter pipelines	env:* 💌								
PIPELINE			EXECUTIONS	FAILURES	FAILURE %	< MEDIAN	MEDIAN CHANGE	> LAST BUILD	DURATION
Q ravinamestry/x22177264_ravina_me	stry_research_pro Build Sonarqu	ube: x22177264_r	1	0	0%	1 min 13 s		SUCCESS	1 min 13 s
O ravinamestry/x22177264_ravina_me	stry_research_pro Deploy STAGE	E: x22177264_ravi	1	0	0%	37.0 s		SUCCESS	37.0 s

Figure 19: Datadog Pipeline Visibility

3.8.2 Set up for Cloud Security Posture Management (CSPM)

- Datadog Integrations tile clicked on AWS and selected AWS region and Datadog API Key.
- Enabled Cloud Security Posture Management to scan the cloud environment, hosts, and containers.
- Clicked on Launch CloudFormation Template and created stack.
- After the stack is created, on the AWS integration tile in Datadog and clicked Ready.

URL: <u>Setting Up CSPM (datadoghq.com)</u>

	aws	Azure	
0	Configure y	bur AWS accounts	
	How to set up P	isture Management for AWS?	
	Configuring Post	ure Management for AWS is a 2-step process. You first need to add necessary permissions resource collection. Then you will be able to enable the collection of these resource	25.
	To apply EC2 me	ric collection filters to CSPM resource collection, contact Support 🗹.	
	Showing 1-1 of	account connected to Datadog	Q Search for an account
	ACCOUNT	TAGS	RESOURCE COLLECTION (CSPM)
	367929450121	aws_account:367929450121	Collect resources

Figure 20: Datadog CSPM Configuration

- Enabled Collect Resources toggle for AWS Resource in Figure 20 in Security>Setup.
- Enabled Cloud Security Posture Management Collection to enable resource collection for CSPM in AWS Integration tile.
- Figure 22 shows the Cloud Security Posture with misconfigured resources and posture score.

15	
16	D0_API_KEY=\${{ secrets.Datadog_API_Key }} D0_SITE="datadoghq.eu" bash -c "\$(curl -L https://s3.amazonaws.com/dd-agent/scripts/install_script_agent7.sh)"
17	sudo apt-get update -y
18	sudo apt-get install apt-transport-https curl gnupg -y
19	sudo sh -c "echo 'deb [signed-by=/usr/share/keyrings/datadog-archive-keyring.gpg] https://apt.datadoghq.com/ stable 7' > /etc/apt/sources.list.d/datadog.list"
20	sudo touch /usr/share/keyrings/datadog-archive-keyring.gpg -y
21	sudo chmod a+r /usr/share/keyrings/datadog-archive-keyring.gpg -y
22	curl https://keys.datadoghq.com/DATADOG_APT_KEY_CURRENT.public sudo gpgno-default-keyringkeyring /usr/share/keyrings/datadog-archive-keyring.gpgimportbatch
23	curl https://keys.datadoghq.com/DATADOG_APT_KEY_C0962C7D.public sudo gpgno-default-keyringkeyring /usr/share/keyrings/datadog-archive-keyring.gpgimportbatch
24	curl https://keys.datadoghq.com/DATADOG_APT_KEY_F14F620E.public sudo gpgno-default-keyringkeyring /usr/share/keyrings/datadog-archive-keyring.gpgimportbatch
25	curl https://keys.datadoghq.com/DATADOG_APT_KEY_382E94DE.public sudo gpgno-default-keyringkeyring /usr/share/keyrings/datadog-archive-keyring.gpgimportbatch
26	sudo cp -a /usr/share/keyrings/datadog-archive-keyring.gpg /etc/apt/trusted.gpg.d/ -y
27	sudo apt-get update -y
28	sudo apt-get install datadog-agent datadog-signing-keys -y
	sudo sh -c "sed 's/api_key:.*/api_key: \${{ secrets.Datadog_API_Key }}/' /etc/datadog-agent/datadog.yaml > /etc/datadog-agent/datadog.yaml"
	sudo -u dd-agent datadog-agent import /etc/dd-agent /etc/datadog-agent -y
	sudo sh -c "chown dd-agent:dd-agent /etc/datadog-agent/datadog.yaml && chmod 640 /etc/datadog-agent/datadog.yaml"
	sudo systemctl restart datadog-agent.service

Figure 21: Datadog agent installation in terraform

	CSM Beerview Posture Management Workload Security Resource Catalog NEW					
	Summary Findings Si	gnals Compliance Rules				
DATADOG	P	OSTURE MANAGEMENT				
Q, Go to		Harden vour cloud resources against	misconfigurations			
💾 Watchdog						
🔲 Service Mgmt 🔸		Resources scanned	Misconfigured Resources	Posture Score		
📥 Dashboards 🔸			7.00	50 6404		
🔮 Infrastructure 🔸		29.0	7.00	58.61% OA1PT N		
			With CRITICAL or HIGH findings			
(″) Metrics →		View All Resources >	View Resources >	How is it Calculated? [5]		
💤 Integrations >						
🐺 АРМ →						
പ ⇒		Overview				
🔊 Notebooks >		Pircennig data by. evaluation, fair X				
ற Logs →						
Security →		Resource Inventory	Evaluation Trend	eventy Trend		
्रेः UX Monitoring ।		and_ublc_victure (117576) and_ublc_function (122576) ansutclut_function (122576) and_ublc_victure (117576) ansutclut_function (122576) ansutclut_function (122576) anst_ublc_victure (117576) ansutclut_function (122576) ansutclut_function (122576) anst_ublc_victure (117576) ansutclut_function (122576) ansutclut_function (122576) anst_ublc_victure (117576) ansutclut_function (122576) ansutclut_function (122576) ansutclut_function (122576) ansutclut_function (122576) ansutclut_function (122576)	PALL 16 PASS 23 SRN 20	CHITCAL HIGH LOW COW 1 (4%) 6 (28%) 11 (55%) 2 (%) 10		
Contact Support		Cloud Accounts	Detection Bulles	O Saveh		
🛛 Help 💛		Croud Accounts	Detection Rules	C search		
ວິລິ Invite Users		ACCOUNT PROVIDER SCANNED RE	SOURCES V FAILED RESOURCES ISSUE	↑ SEVERITY J ² FAILED RESOURCES		
X22177264@st, NCI		367929450121 aws	29 16 (55.17%) MFA is enabled for the "root" account	CRITICAL 1 (100.00%)		

Figure 22: Datadog Posture Management

3.8.3 Set up for Datadog Agent

• Got the steps for installing Datadog agent from the URL mentioned in Table 1 and added the steps in the terraform

terraform/installdocker.sh file shown in Figure 21.

• Below command shows the status of the Datadog agent running for the host.

sudo service datadog-agent status

3.9 Super-Linter

- Entered the code in Figure 23 in the .GitHub/workflows/build.yml workflow from the URL mentioned below.
- GitHub Actions will automatically run the Super-Linter workflow whenever there is a pull request or push event to the specified branches.

URL: https://github.com/marketplace/actions/super-linter



Figure 23: Super-Linter code in build

4 Configuring the Pipeline

• Figure 24 illustrates the summary of CI-CD pipeline with integrated Security tools on various stages of the pipeline designed for the research paper explained in sections.



Figure 24: CI-CD Pipeline with Integrated Security Tools

4.1 Infrastructure Deployment Workflow

• Below workflow in Figure 25 is created for infrastructure development, GitHub actions will trigger terraform_create.yml workflow when there are changes in infrastructure requirements from terraform terraform/main.tf path.



Figure 25: Infrastructure Deployment Workflow

4.1.1 Pull requests steps:

- Figure 26 shows pull request steps and merge to develop and merge to master steps in terraform plan and apply execution.
- On pull request, terraform initiates the directory which has terraform Configuration files and validates the Configuration.
- Terraform Plan shows the plan which has the actions Terraform will take in order.

• Terraform plan status returns whether a plan was successfully generated or not.

4.1.2 Merge to master steps:

- Terraform runs the Apply when Pull request is merged into master branch.
- Terraform Apply the plan and creates the infrastructure according to the plan.
- It retrieves the public IPs of instances and saves them in GitHub secrets shown in Figure 27.



Figure 26: Terraform Plan and Apply Execution

4.2 Application Deployment Workflow

- Develop branch from GitHub repo consists of code change which is to be added to the development and then to Staging environments.
- When the Pull request is generated on Develop branch, GitHub actions will trigger build.yml workflow on develop branch for webapp illustrated in Figure 28.

A X22177264_RAVINA_MESTRY_RESEARCH_PROJECT_DEV_PUBLIC_IP	Updated 4 hours ago	Ĵ
A X22177264_RAVINA_MESTRY_RESEARCH_PROJECT_PROD_PUBLIC_IP	Updated 4 hours ago	1 D
A X22177264_RAVINA_MESTRY_RESEARCH_PROJECT_SONARQUBE_PUBLIC_ IP	Updated 3 hours ago	Ĵ
A X22177264_RAVINA_MESTRY_RESEARCH_PROJECT_STAGE_PUBLIC_IP	Updated 4 hours ago	Ø Ů
A X22177264_RAVINA_MESTRY_RESEARCH_PROJECT_ZAP_PUBLIC_IP	Updated 3 hours ago	2 0

Figure 27: GitHub Secrets for Public IP



Figure 28: Application Deployment Workflow

- It builds and push the web app and Docker Hub image with tag 'x22177264_ravina_mestry/project:ncirl_cloud_research_webapp_x22177264_ravina __mestry'.
- When the develop branch is merged into develop branch shown in Figure 28, GitHub actions will trigger deploy-dev.yml workflow on develop branch. It deploys the application in development environment from GitHub Action secrets x22177264_ravina_mestry_research_project_dev_PUBLIC_IP'.
- It gets docker-compose.yml file and removes existing Docker container and image, gets Login for Docker, and runs Docker to deploy the application in development environment from GitHub Action secrets x22177264_ravina_mestry_research_project_dev_PUBLIC_IP'.

- When the develop branch is merged into master branch, GitHub actions will trigger deploy-stage.yml workflow on master branch.
- When GitHub actions is manually triggered with input PROD in the workflow in Figure 29, removes existing Docker container and image and the deploy.yml workflow and deploys the application in Production environment from GitHub Action secrets x22177264_ravina_mestry_research _project_prod PUBLIC IP'.

Run v	vorkflow -
Use workflow from Branch: master *	
Environment [DEV/STAGE/PROD] *	
Run workflow	

Figure 29: GitHub Action Deploy workflow

4.3 Destroy Infra Workflow

• The destroy infra workflow destroys no longer needed infrastructure. Resources like Dev or Stage are destroyed and created when needed. The Terraform destroy command terminates the resources in workspace. Figure 30 shows the terraform_destroy.yml execution destroying the dev, stage, and prod AWS EC2 instances.

terraform_destroy.yml terraform_destroy_sonarqube.yml terraform_destroy_zap.yml

~ Ø	Terraform Destroy
	Changes to Outputs:
257	<pre>- x22177264_ravina_mestry_research_project_dev_public_ip = "100.24.236.187" -> null</pre>
	<pre>- x22177264_ravina_mestry_research_project_prod_public_ip = "54.146.82.165" -> null</pre>
	<pre>- x22177264_ravina_mestry_research_project_stage_public_ip = "54.145.239.6" -> null</pre>
	aws_instance.x22177264_ravina_mestry_research_project_stage: Destroying [id=i-08ef33cd2b66bfbcc]
	aws_instance.x22177264_ravina_mestry_research_project_prod: Destroying [id=i-0fdc7728205f1f619]
	aws_instance.x22177264_ravina_mestry_research_project_dev: Destroying [id=i-038a1fdee2ab88667]
264	<pre>aws_instance.x22177264_ravina_mestry_research_project_stage: Still destroying [10s elapsed]</pre>
	<pre>aws_instance.x22177264_ravina_mestry_research_project_dev: Still destroying [10s elapsed]</pre>
	<pre>aws_instance.x22177264_ravina_mestry_research_project_prod: Still destroying [10s elapsed]</pre>
267	<pre>aws_instance.x22177264_ravina_mestry_research_project_prod: Still destroying [20s elapsed]</pre>
	<pre>aws_instance.x22177264_ravina_mestry_research_project_stage: Still destroying [20s elapsed]</pre>
	aws_instance.x22177264_ravina_mestry_research_project_dev: Still destroying [20s elapsed]
270	aws_instance.x22177264_ravina_mestry_research_project_prod: Destruction complete after 30s
271	aws_instance.x22177264_ravina_mestry_research_project_dev: Still destroying [30s elapsed]
272	<pre>aws_instance.x22177264_ravina_mestry_research_project_stage: Still destroying [30s elapsed]</pre>
273	aws_instance.x22177264_ravina_mestry_research_project_stage: Destruction complete after 40s
274	aws_instance.x22177264_ravina_mestry_research_project_dev: Destruction complete after 40s
275	
276	Apply complete! Resources: 0 added, 0 changed, 3 destroyed.
277	

Figure 30: Destroy Infrastructure Workflow