

Configuration manual

MSc Research Project Cloud Computing

Ram Prakash Student ID: x22132236

School of Computing National College of Ireland

Supervisor: Rashie

Rashid Mijumbi

National College of Ireland Project Submission Sheet School of Computing



Student Name:	Ram Prakash			
Student ID:	x22132236			
Programme:	Cloud Computing			
Year:	2023			
Module:	MSc Research Project			
Supervisor:	Rashid Mijumbi			
Submission Due Date:	14-12-2023			
Project Title:	Benchmarking Container Orchestration Tools on Applicati			
	Performance in the Cloud			
Page Count:	6			

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:	
Date:	31st January 2024

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

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

Ram Prakash x22132236

1 Section 1

1.1 Setting Up the Development Environment

Install Visual Studio Code:

- 1. Download and install Visual Studio Code from the official website.
- 2. Install necessary extensions for React development, such as ESLint and Prettier.

Building the React App:

- Use npx create-react-app my-react-app to create a new React application.
- Develop the application using VS Code.



Figure 1: Visual Studio

2 Section 2

2.1 Containerization and AWS ECR

Install Docker:

- Download and install Docker Desktop from Docker's official website.
- Download the exe file for Windows from the below link
- https://www.docker.com/products/docker-desktop/
- Download the.deb or.rpm file from the below link followed by installation steps in the official documentation for docker
- https://docs.docker.com/desktop/install/linux-install/

Writing a Dockerfile:

- Create a Dockerfile in your project's root directory.
- Define the container image, e.g., use node:alpine, copy files, install dependencies, and set the start command.



Figure 2: Dockerfile

•

Building and Pushing to ECR:

- Build the Docker image using docker build -t my-react-app .
- Tag and push the image to AWS ECR.

R	i-0ae64e4e8f1b0f785	⊘Running @ Q	t2.micro	() Initializing	No alarms 🕂	us-east-1d	ec2-54-225-21-90.com	54.225.21.90
Ū.	i-028f9dc8f58f231e1	⊘ Running @ Q	t2.micro	 Initializing 	No alarms +	us-east-1d	ec2-54-227-174-50.co	54.227.174.5
	i-057ce046c3d798108	📿 Running 🕘 🤤	t2.micro	 Initializing 	No alarms +	us-east-1d	ec2-3-93-70-212.comp	3.93.70.212

Figure 3: Docker Swarm Nodes

3 Section 3

3.1 Docker Swarm Setup

Creating EC2 Instances:

- Create three EC2 t2.micro instances in AWS.
- SSH into each instance for setup.

Initializing Docker Swarm:

- Initialize Docker Swarm on the master node.
- Join the worker nodes using the provided token.

Deploying the Service:

• Create a Docker Swarm service to deploy the React SPA.

1 ite stack	Create stack					
2 ify stack details	Prerequisite - Prepare template					
3 Figure stack options	Prepare template Every stack is based on a template. A template is a JSON or YAML file th	Prepare template Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.				
igure stack options	 Template is ready 	 Use a sample template 	 Create template in Designer 			
2W						
*	Specify template A template is a 30% or VAHL file that describes your stack's resources a Template source Selecting a template generates an Amazon 51 URL where it will be ston	ed				
*	Specify template A template is a JSON or VAML file that describes your stack's resources a Template source Selecting a template generates an Amazon S1 URL where it will be ston P Amazon S3 URL Provide an Amazon S3 URL to your template. Amazon S3 URL	ed. O Upload a template file Upload your template directly to the console.	Sync from Git - new Sync a template from your Git repository.			
ew •	Specify template A template is a 350% or VAML file that describes your stack's resource a Template source Selecting a template generates an Amazon 53 URL where it will be store Amazon 53 URL Provide an Amazon 53 URL to your template. Amazon 53 URL https://	ed O Upload a template file Upload your template directly to the console.	Sync from Git - new Sync a template from your Git repository.			
rw.	Specify template A template is a JSON or VAML file that describes your stack's resources a Template source Selecting a template generates an Amazon S5 URL where it will be ston Mazzon S3 URL Powide an Amazon S1 URL to your template. Intersci S1 URL Amazon S1 template URL	ed. O Upload a template file Upload your template directly to the console.	Sync from Git - new Sync a template from your Git repository.			

Figure 4: docker swarm

4 Section 4

4.1 Kubernetes (EKS) Setup

Install AWS CLI and eksctl:

• Follow the official guides to install the AWS CLI and eksctl.

Creating an EKS Cluster:

• Use eksctl to create an EKS cluster with the required configuration.

Deploying to EKS:

• Apply deployment and service configurations using kubectl.

YS C:\Users\jojot\Documents →	\RIC\Main\Ram	\benchmark\orchest	ration-benchmark-test> kubectl get service my-orchestration-test-a Add to cha
IAME 5) AGE	ТҮРЕ	CLUSTER-IP	EXTERNAL-IP
y-orchestration-test-app 90/TCP 4m54s	LoadBalancer	10.100.240.93	a6eb4ad106b234b17810de06b8c2d4e2-814482681.us-east-1_elb.amazonaw
YS C:\Users\jojot\Documents	\RIC\Main\Ram	\benchmark\orchest	ration-benchmark-test> kubectl get deployment
IAME	READY UP-T	O-DATE AVAILABLE	AGE
v-orchestration-test-app	0/3 3	0	7m1s

Figure 5: EKS cluster

5 Section 5

5.1 Performance Testing

Installing Ubuntu and Apache Bench:

- Install Ubuntu either through WSL or on a desktop.
- Install Apache Bench for performance testing.

Executing Performance Tests:

• Use Apache Bench to simulate traffic to your application.

5.2 Monitoring and Analysis

Using AWS CloudWatch:

• Setup CloudWatch in AWS for monitoring.

Analyzing Performance Data:

- Collect data from Apache Bench and CloudWatch for analysis.
- Document and analyze the performance metrics.

Server Software: Server Hostname: nstalk.com Server Port:		nginx orchestrationdockerswarm.us-west-2.elasticbea 80						
Document Path: Document Length:		/ 5260 byt	/ 5260 bytes					
Concurrency Level: Time taken for tests: Complete requests: Failed requests: Total transferred: HTML transferred: Requests per second: Time per request: Time per request:		1 36.212 s 100 0 555300 b 526000 b 2.76 [#, 362.123 362.123	1 36.212 seconds 100 0 555300 bytes 526000 bytes 2.76 [#/sec] (mean) 362.123 [ms] (mean, across all concurrent req					
Transfer rate	:	14.98 [H	<bytes <="" td=""><td>'sec] received</td></bytes>	'sec] received				
Connection Ti	mes (ms)	F . / 13						
Connect: Processing: Waiting: Total:	min mear 172 179 176 183 175 182 348 362	17.5 11.3 11.8 11.8 25.4	nedian 176 180 179 357	max 312 256 256 521				
Percentage of 50% 357 66% 360 75% 363 80% 364 90% 366 95% 375 98% 492 99% 521 100% 521	the reque	equest)	ed with	in a certain time (ms)				

Figure 6: apache Beam Load testing



Figure 7: Cloud watch