

# Configuration Manual

MSc Research Project MSc Cloud Computing

Nikita Patel Student ID: 21224811

School of Computing National College of Ireland

Supervisor: Sean Heeney

#### National College of Ireland MSc





#### of Computing

**Student Name:** Nikita Patel

**Student ID:** 21224811

**Programme:** MSc Cloud Computing **Year:** 2022-2023

**Module:** MSc Research Project

**Lecturer:** Sean Heeney

Submission Due

**Date:** 18-09-2023

**Project Title:** Webhook Driven Cross Platform Docker Image

Transfer: Achieving AWS-Azure Interoperability

Word Count: 754 Page Count: 7

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:** Nikita Patel

**Date:** 17-09-2023

#### 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**

Nikita Patel Student ID: 21224811

# 1 Pre-requisites and access required to cloud services

We should have active accounts with both Azure and AWS to access the respective cloud services. Sign up for free-tier accounts or we can use existing accounts if available.

### **Azure Account Pre-requisites:**

- ♦ Azure Subscription: Ensure we have an active Azure subscription to access Azure services.
- ♦ Azure Functions: Familiarize with Azure Functions, Microsoft's serverless compute service.

Monitoring	Logs	Capabilities	Notifications	Recommendations		
ор						
	RIC-WEB-APP					
ng model						
Container Image			ricregistry.azurecr.io/nikitapatel29/ric_project:36			
	op ng model	op ng model	RIC-WEI	RIC-WEB-APP  Roman Container		

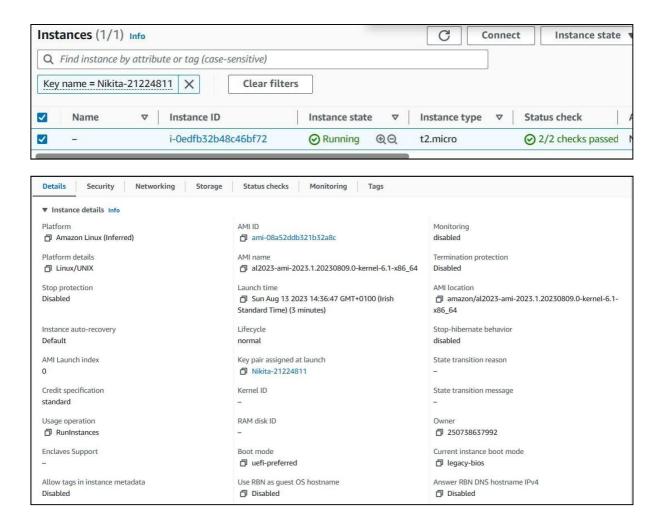
#### **AWS Account Pre-requisites:**

- ♦ AWS Account: Ensure to have an active AWS account with necessary permissions to access AWS services.
- ♦ AWS Lambda: Get acquainted with AWS Lambda, Amazon's serverless compute service².
- ♦ Amazon EC2: Set up an Amazon EC2 to host Docker container images.

1

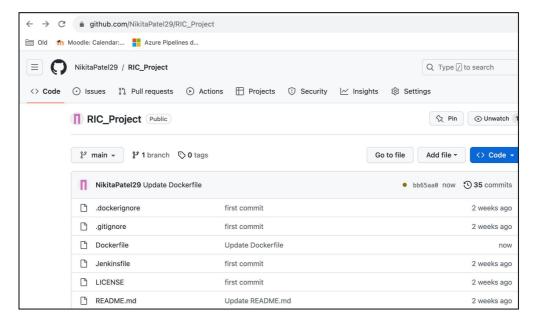
<sup>1</sup> https://learn.microsoft.com/en-us/azure/devops/service-hooks/services/webhooks?view=azure-devops

<sup>&</sup>lt;sup>2</sup> https://docs.aws.amazon.com/ec2/index.html



#### **Application Code Repository**:

♦ The application's source code stored in a version control repository, in a Git accessible from Azure DevOps

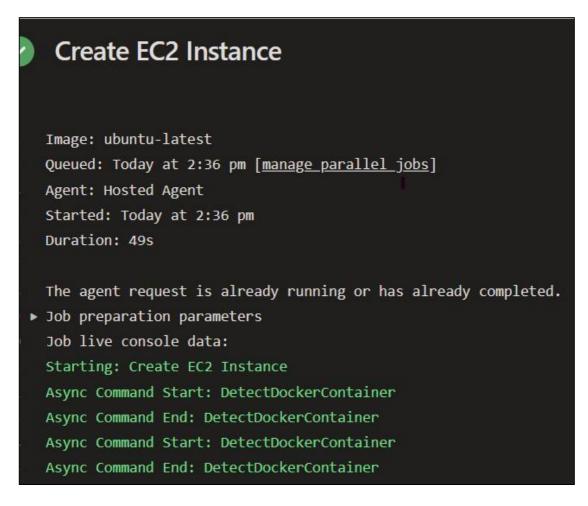


#### **Containerization Skills:**

♦ The basics of Docker containerization and how to create Docker images for your applications.

#### **Docker Image and Dockerfile:**

♦ Ensure we have a Docker image built using a Dockerfile that specifies the application's runtime and dependencies<sup>3</sup>.



By fulfilling these pre-requisites and having access to the required cloud services, we are well-equipped to embark on exploration of integrating serverless computing with container deployment across Azure and AWS environments.

## 2 Configuration Steps:

#### **Setting Up Azure:**

♦ Sign up for an Azure account.

<sup>&</sup>lt;sup>3</sup> https://docs.docker.com.zh.xy2401.com/v17.12/

♦ Connect the GitHub repository to Azure Pipelines to enable automated deployment.

#### **Configuring AWS EC2:**

- ♦ Create an Amazon EC2 cluster within the AWS account. This cluster will manage the deployment of Docker containers.
- ♦ Set up an Amazon EC2 repository to store the Docker images.
- ♦ Create an EC2 task that includes the specifications for the containers, such as image repository details and resource allocation.

#### **Creating a Webhook:**

- ❖ Inside the GitHub repository settings, navigate to Webhooks.
- ♦ Create a new webhook that will trigger the Azure code changes that are pushed to the repository.
- ♦ Configure the payload URL to point to your Azure endpoint.

#### **Webhook Implementation:**

- ♦ In the Azure, integrate the webhook's payload data.
- ♦ Customize to initiate Docker image builds automatically whenever the webhook triggers a code commit.

#### **Image Transfer to AWS:**

- ♦ After successful Docker image creation in the Azure the scripts will authenticate the AWS EC2 repository.
- → Push the newly built Docker image from Azure to the AWS EC2 repository.

```
Starting: Create EC2 Instance
            : Command line
Description : Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
             : Microsoft Corporation
Author
             : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/command-line
Generating script.
                       ===== Starting Command Output ======
/usr/bin/bash --noprofile --norc /home/vsts/work/_temp/7d97bd59-6be3-44a0-96dd-b27b4e34baeb.sh
Collecting awscli
 Obtaining dependency information for awscli from <a href="https://files.pythonhosted.org/packages/58/89/828753db4c6d8079d059a8a4a7a">https://files.pythonhosted.org/packages/58/89/828753db4c6d8079d059a8a4a7a</a>
 Downloading awscli-1.29.25-py3-none-any.whl.metadata (11 kB)
Collecting botocore==1.31.25 (from awscli)
 Obtaining dependency information for botocore==1.31.25 from <a href="https://files.pythonhosted.org/packages/6b/2f/74967de70d1fc0fb">https://files.pythonhosted.org/packages/6b/2f/74967de70d1fc0fb</a>
  Downloading botocore-1.31.25-py3-none-any.whl.metadata (5.9 kB)
Collecting docutils<0.17.>=0.10 (from awscli)
 Downloading docutils-0.16-py2.py3-none-any.whl (548 kB)
                                              —— 548.2/548.2 kB 5.1 MB/s eta 0:00:00
Collecting s3transfer<0.7.0,>=0.6.0 (from awscli)
 Downloading s3transfer-0.6.1-py3-none-any.wh1 (79 kB)
                                                 - 79.8/79.8 kB 6.8 MB/s eta 0:00:00
Collecting PyYAML<6.1,>=3.10 (from awscli)
 Obtaining dependency information for PyYAML<6.1,>=3.10 from https://files.pythonhosted.org/packages/7b/5e/efd033ab7199a0b2
 Downloading\ PyYAML-6.0.1-cp311-cp311-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata\ (2.1\ kB)
 Downloading colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Collecting rsa<4.8,>=3.1.2 (from awscli)
```

#### **AWS EC2 Deployment:**

- ♦ Configure an EC2 service within your EC2 cluster. Define the desired number of tasks and assign them to the task definition you created earlier.
- ♦ Set up the service to pull the Docker image from your AWS EC2 repository.
- ♦ Launch the EC2 service, allowing it to deploy containers based on the image you pushed.

```
Collecting pyasn1>=0.1.3 (from rsa<4.8,>=3.1.2->awscli)
      Downloading pyasn1-0.5.0-py2.py3-none-any.whl (83 kB)
                                                - 83.9/83.9 kB 8.3 MB/s eta 0:00:00
43 Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->botocore==1.31.25->awscli)
    Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Downloading awscli-1.29.25-py3-none-any.whl (4.2 MB)
                                              - 4.2/4.2 MB 12.0 MB/s eta 0:00:00
47 Downloading botocore-1.31.25-py3-none-any.whl (11.1 MB)
                                              - 11.1/11.1 MB 33.9 MB/s eta 0:00:00
   Downloading PyYAML-6.0.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (757 kB) ⊖
                                              - 757.7/757.7 kB 57.8 MB/s eta 0:00:00
51 Downloading urllib3-1.26.16-py2.py3-none-any.whl (143 kB)
                                             — 143.1/143.1 kB 32.7 MB/s eta 0:00:00
    Installing collected packages: urllib3, six, PyYAML, pyasn1, jmespath, docutils, colorama, rsa, python-dateutil, botocore, s
    Successfully installed PyYAML-6.0.1 awscli-1.29.25 botocore-1.31.25 colorama-0.4.4 docutils-0.16 jmespath-1.0.1 pyasn1-0.5.0
55 Finishing: Create EC2 Instance
```

### **Verification:**

♦ Access the deployed application on AWS EC2 to verify that cross-platform interoperability has been achieved.

### References

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.