

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
Project Submission Sheet School
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.

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: 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)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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.
- ✧ Azure Container Registry: Set up an Azure Container Registry to store Docker container images.¹



The screenshot shows the 'Properties' tab of an Azure Web app. The 'Name' is 'RIC-WEB-APP', the 'Publishing model' is 'Container', and the 'Container Image' is 'ricregistry.azurecr.io/nikitapatel29/ric_project:36'. Other tabs visible are 'Monitoring', 'Logs', 'Capabilities', 'Notifications', and 'Recommendations'.

Properties	Monitoring	Logs	Capabilities	Notifications	Recommendations
 Web app					
Name					
Publishing model					
Container Image					

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

² <https://docs.aws.amazon.com/ec2/index.html>

Instances (1/1) Info Refresh Connect Instance state

Find instance by attribute or tag (case-sensitive)

Key name = Nikita-21224811 Clear filters

Name	Instance ID	Instance state	Instance type	Status check
-	i-0edfb32b48c46bf72	Running	t2.micro	2/2 checks passed

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance details info

Platform Amazon Linux (Inferred)	AMI ID ami-08a52ddb321b32a8c	Monitoring disabled
Platform details Linux/UNIX	AMI name al2023-ami-2023.1.20230809.0-kernel-6.1-x86_64	Termination protection Disabled
Stop protection Disabled	Launch time Sun Aug 13 2023 14:36:47 GMT+0100 (Irish Standard Time) (3 minutes)	AMI location amazon/al2023-ami-2023.1.20230809.0-kernel-6.1-x86_64
Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 0	Key pair assigned at launch Nikita-21224811	State transition reason -
Credit specification standard	Kernel ID -	State transition message -
Usage operation RunInstances	RAM disk ID -	Owner 250738637992
Enclaves Support -	Boot mode uefi-preferred	Current instance boot mode legacy-bios
Allow tags in instance metadata Disabled	Use RBN as guest OS hostname Disabled	Answer RBN DNS hostname IPv4 Disabled

Application Code Repository:

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

github.com/NikitaPatel29/RIC_Project

NikitaPatel29 / RIC_Project

Code | Issues | Pull requests | Actions | Projects | Security | Insights | Settings

RIC_Project Public

main | 1 branch | 0 tags

NikitaPatel29 Update Dockerfile • bb65aa0 now 35 commits

.dockerignore	first commit	2 weeks ago
.gitignore	first commit	2 weeks ago
Dockerfile	Update Dockerfile	now
Jenkinsfile	first commit	2 weeks ago
LICENSE	first commit	2 weeks ago
README.md	Update README.md	2 weeks ago

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 dependencies³.

```

Create EC2 Instance

Image: ubuntu-latest
Queued: Today at 2:36 pm [manage_parallel_jobs]
Agent: Hosted Agent
Started: Today at 2:36 pm
Duration: 49s

The agent request is already running or has already completed.
▶ Job preparation parameters
Job live console data:
Starting: Create EC2 Instance
Async Command Start: DetectDockerContainer
Async Command End: DetectDockerContainer
Async Command Start: DetectDockerContainer
Async Command End: DetectDockerContainer

```

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.

³ <https://docs.docker.com.zh.xnxy2401.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
=====
Task          : Command line
Description   : Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
Version      : 2.212.0
Author       : Microsoft Corporation
Help        : 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 https://files.pythonhosted.org/packages/58/89/828753db4c6d8079d059a8a4a7a
  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 https://files.pythonhosted.org/packages/6b/2f/74967de70d1fc0fb
  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.whl (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)
Collecting colorama<0.4.5,>=0.2.5 (from awscli)
  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.

```
40 Collecting pyasn1>=0.1.3 (from rsa<4.8,>=3.1.2->awscli)
41   Downloading pyasn1-0.5.0-py2.py3-none-any.whl (83 kB)
42     _____ 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)
44   Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
45   Downloading awscli-1.29.25-py3-none-any.whl (4.2 MB)
46     _____ 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)
48     _____ 11.1/11.1 MB 33.9 MB/s eta 0:00:00
49   Downloading PyYAML-6.0.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (757 kB)
50     _____ 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)
52     _____ 143.1/143.1 kB 32.7 MB/s eta 0:00:00
53 Installing collected packages: urllib3, six, PyYAML, pyasn1, jmespath, docutils, colorama, rsa, python-dateutil, botocore, s
54 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.