

# Configuration Manual

MSc Research Project  
MSC CLOUD COMPUTING

**MANISHA PRASAD**

Student ID: x21231222

School of Computing  
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland  
Project Submission Sheet  
School of Computing



<b>Student Name:</b>	MANISHA PRASAD
<b>Student ID:</b>	x21231222
<b>Programme:</b>	MSC CLOUD COMPUTING
<b>Year:</b>	2023
<b>Module:</b>	MSc Research Project
<b>Supervisor:</b>	Vikas Sahni
<b>Submission Due Date:</b>	14/12/2023
<b>Project Title:</b>	Configuration Manual
<b>Word Count:</b>	580
<b>Page Count:</b>	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.

<b>Signature:</b>	
<b>Date:</b>	14th December 2023

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission</b> , to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project</b> , 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.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

MANISHA PRASAD

x21231222

## 1 Introduction

### 1.1 Purpose Of The Document

This document discuss tools and services required for the deployment of proposed methodology along with configuration manual that present complete guide for deployment.

## 2 Document Overview

### 2.1 Objective

This study aims to demonstrate how such integration may significantly improve illness prediction accuracy, optimise healthcare application delivery, and eventually contribute to the establishment of a more efficient and refined healthcare infrastructure.

### 2.2 Tools

Amazon Sagemaker, Elastic beanstalk , codepipeline ,Cloud 9, Github

### 2.3 Github

In the proposed methodology , Github is used for the source control repository to store the flask based webapplication.

### 2.4 Sagemaker

In this notebook instance of AWS SageMaker is used to prepare and process the data along with training the model. This notebook instance of AWS sagemake is integrated with python jupyter notebook.

### 2.5 Cloud 9

In this project Cloud 9 IDE is used to run the terraform script for provisioning the AWS ElasticBeanstalk Environment along with Beanstalk Application.

### 2.6 ElasticBeanstalk

Now in this, it was used to host the Flaskbased application which is stored in Github Repository

## 2.7 CodePipeline

Now Aws codepipeline is used to automate deployment of web application on elasticbeanstalk environment, It will fetch webapplication source code from Github repository and push it on elasticbeanstalk environment

## 3 Configuration:

### 3.1 Webapplication Source Code and Terraform Script

- Please refer the zip file submitted in ICT Solution Artefact

### 3.2 Dataset URL

<https://archive.ics.uci.edu/dataset/45/heart+disease>

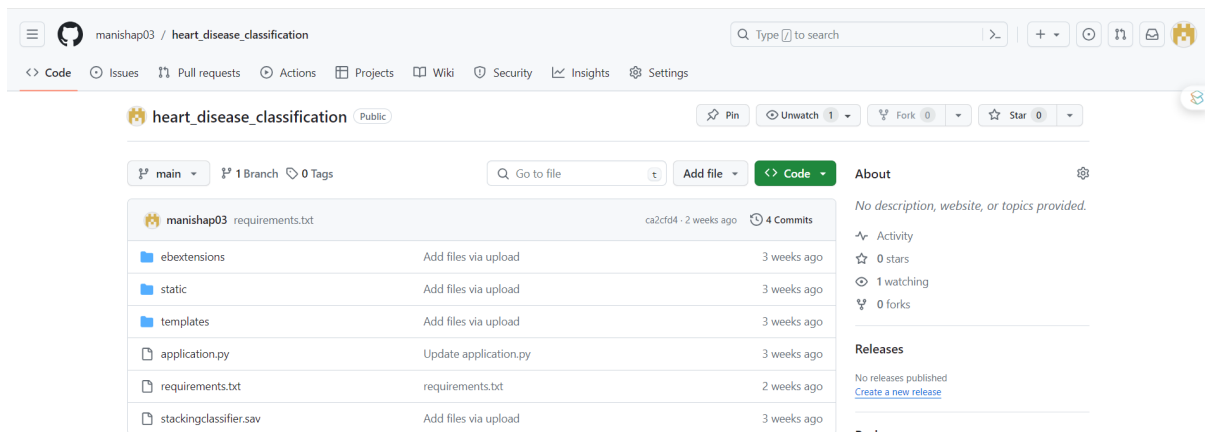
### 3.3 Github :

Step 1: Login to Github account or create new one if you don't have

Step 2 : Create Repository

Step 3: Upload the source code of webapplication by using Git CLI Commands.

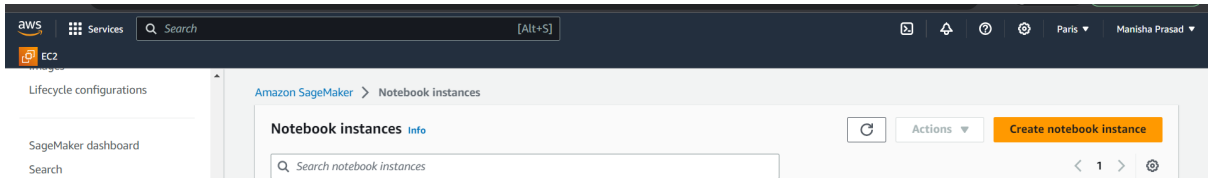
Note – On successful upload the github repository URL will appear that require in the configuration of the AWS codepipeline



### 3.4 Sagemaker

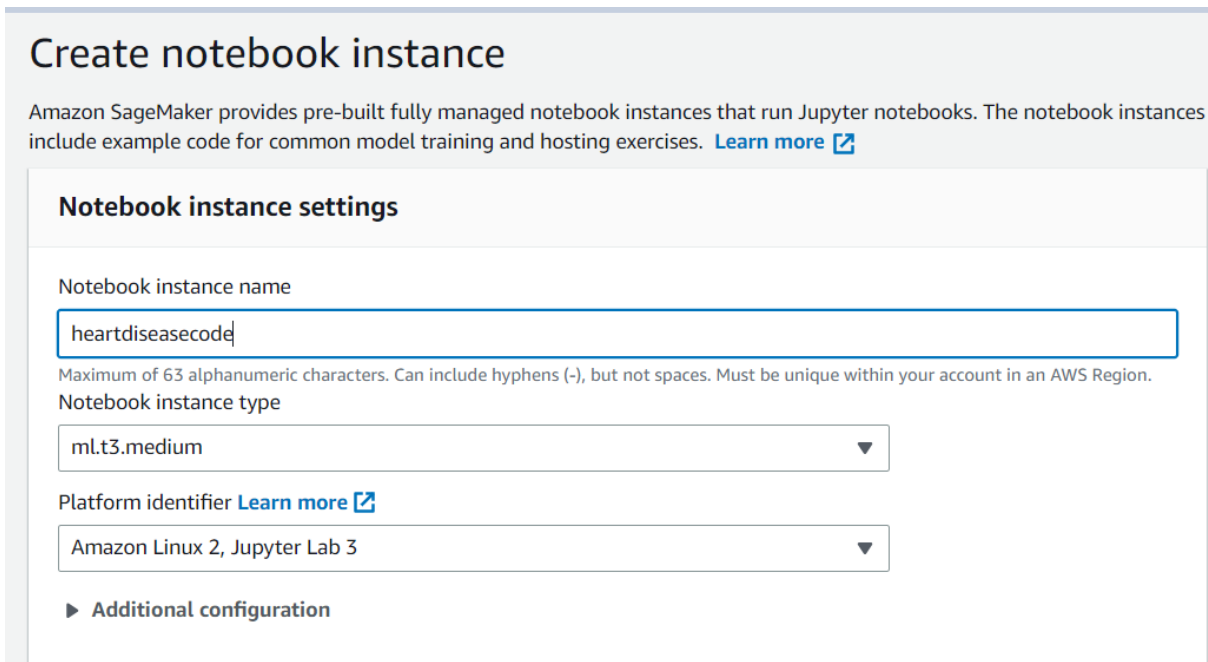
Step 1 : Login to AWS PORTAL.

Step 2 : Navigate to AWS SAGEMAKER

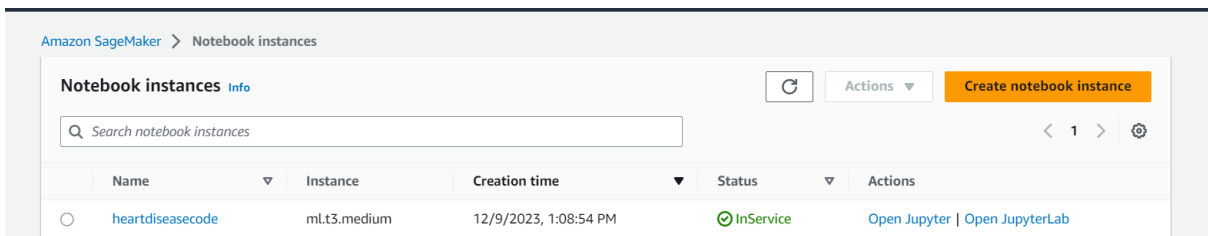


Step 3 : Select Notebook instance option under Notebook

Step 4 : Click on “Create Notebook Instance” and specify the value as shown in the fig



Step 5 : After Successful Creation, open the created notebook instance and click on “open in jupyter lab”



Step 6 : Open lab and upload the data set and code, once upload run the code .

### 3.5 Cloud 9:

Step 1: Create Cloud 9 Environment



## Create environment [Info](#)

### Details

#### Name

Limit of 60 characters, alphanumeric, and unique per user.

#### Description - optional

Limit 200 characters.

#### Environment type [Info](#)

Determines what the Cloud9 IDE will run on.

**New EC2 instance**

Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

**Existing compute**

You have an existing instance or server that you'd like to use.

## New EC2 instance

### Instance type [Info](#)

The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

**t2.micro (1 GiB RAM + 1 vCPU)**

Free-tier eligible. Ideal for educational users and exploration.

**t3.small (2 GiB RAM + 2 vCPU)**

Recommended for small web projects.

**m5.large (8 GiB RAM + 2 vCPU)**

Recommended for production and most general-purpose development.

**Additional instance types**

Explore additional instances to fit your need.

### Platform [Info](#)

This will be installed on your EC2 instance. We recommend Amazon Linux 2.

Amazon Linux 2 ▼

### Timeout

How long Cloud9 can be inactive (no user input) before auto-hibernating. This helps prevent unnecessary charges.

30 minutes ▼

Step 2 : Click on create

**Network settings** [Info](#)

Connection  
How your environment is accessed.

AWS Systems Manager (SSM)  
Accesses environment via SSM without opening inbound ports (no ingress).

Secure Shell (SSH)  
Accesses environment directly via SSH, opens inbound ports.

▶ **VPC settings** [Info](#)

▶ **Tags - optional** [Info](#)  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

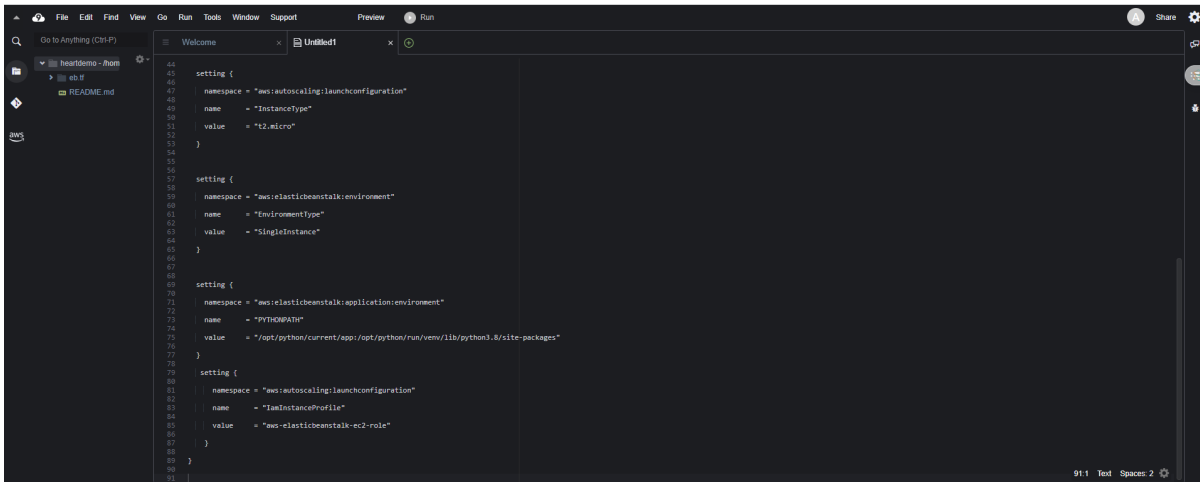
**The following IAM resources will be created in your account**

- **AWSServiceRoleForAWSCloud9** - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

Cancel **Create**

### 3.6 AWS ELASTIC BEANSTALK

Open the created cloud 9 environment and upload and run the terraform script provided in the ICT Solution Artefact

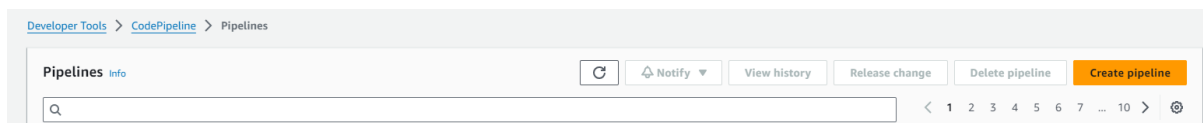
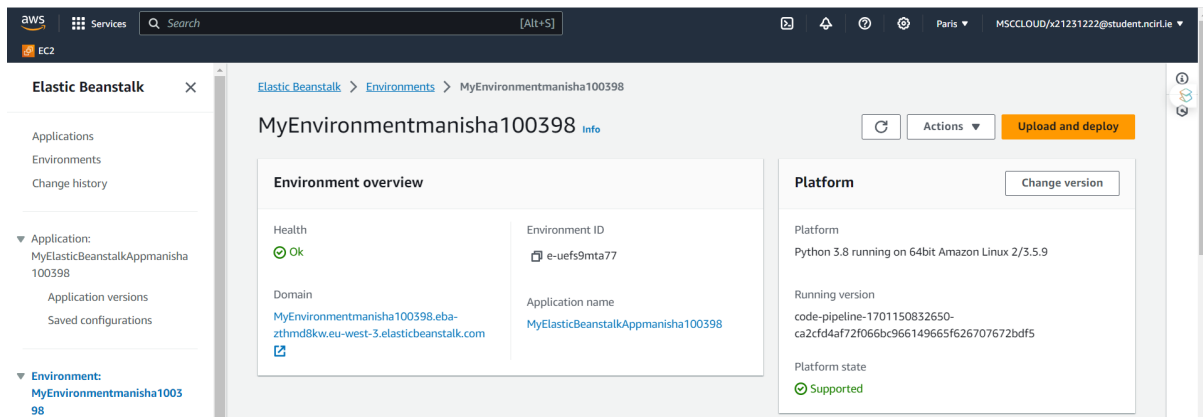


```
44
45
46
47   namespace = "aws:autoscaling:launchconfiguration"
48
49   name      = "InstanceType"
50
51   value     = "t2.micro"
52
53 }
54
55
56
57
58
59
60
61   namespace = "aws:elasticbeanstalk:environment"
62
63   name      = "EnvironmentType"
64
65   value     = "SingleInstance"
66
67 }
68
69
70
71   namespace = "aws:elasticbeanstalk:application:environment"
72
73   name      = "PYTHONPATH"
74
75   value     = "/opt/python/current/app:/opt/python/run/venv/lib/python3.8/site-packages"
76
77 }
78
79
80
81   namespace = "aws:autoscaling:launchconfiguration"
82
83   name      = "IamInstanceProfile"
84
85   value     = "aws-elasticbeanstalk-ec2-role"
86
87 }
88
89 }
90
91
```

After executing above terraform script it will deploy elasticbeanstalk environment

### 3.7 CodePipeline

It is used to retrieve webapplication source code from github repository and push it on elasticbeanstalk environment



Step1 : navigate to AWS Codepipeline

Step 2 : In choose pipeline setting , Specify the name for the pipeline and click next

Step 3: In add source stage, Select the source provider as the github and connect to github account that contain webapplication source code

Note – skip add build stage Step 4 – In add deploy stage , select AWS Elastic beanstalk as source provider , and specify beanstalk environment configuration detail as shown in fig

Now just review and Create Pipeline



## Choose pipeline settings [Info](#)

Step 1 of 5

### Pipeline settings

#### Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

No more than 100 characters

#### Pipeline type

The pipeline type determines the pipeline structure and availability of parameters such as triggers. Pipeline type selection will impact features and pricing. **Which pipeline is right for me?**

 V1 V2

#### Service role

 New service role

Create a service role in your account

 Existing service role

Choose an existing service role from your account

#### Role name

Type your service role name

Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

## Add source stage Info

Step 2 of 5

### Source

#### Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 1) ▼

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connected

✔ You have successfully configured the action with the provider. ✕



#### The GitHub (Version 1) action is not recommended

The selected action uses OAuth apps to access your GitHub repository. This is no longer the recommended method. Instead, choose the GitHub (Version 2) action to access your repository by creating a connection. Connections use GitHub Apps to manage authentication and can be shared with other resources. [Learn more](#)

#### Repository

manishap03/heart\_disease\_classification ✕

#### Branch

main ✕

#### Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

**GitHub webhooks (recommended)**  
Use webhooks in GitHub to automatically start my pipeline when a change occurs

**AWS CodePipeline**  
Use AWS CodePipeline to check periodically for changes

Cancel

Previous

Next

## Add deploy stage Info

Step 4 of 5



### You cannot skip this stage

Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

### Deploy

#### Deploy provider

Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS Elastic Beanstalk

#### Region

Europe (Paris)

#### Application name

Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.

MyElasticBeanstalkAppmanisha100398

#### Environment name

Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.

MyEnvironmentmanisha100398

Cancel

Previous

Next

### Step 5 of 5: Create pipeline

#### Deploy action provider

Deploy action provider

AWS Elastic Beanstalk

ApplicationName

MyElasticBeanstalkAppmanisha100398

EnvironmentName

MyEnvironmentmanisha100398

Cancel

Previous

Create pipeline

Developer Tools > CodePipeline > Pipelines > codepipelinemanisha

### codepipelinemanisha

Pipeline type: V2

[Notify](#) [Edit](#) [Stop execution](#) [Clone pipeline](#) [Release change](#)

**Source** Succeeded  
Pipeline execution ID: 424f5d6c-1949-4b88-8b90-2c73b414d81f

Source  
[GitHub \(Source\)](#)  
Succeeded - 11 days ago  
[View logs](#)

[s32c45a](#) Source: requirements.txt

Disable transition

**Deploy** Succeeded  
Pipeline execution ID: 424f5d6c-1949-4b88-8b90-2c73b414d81f

Deploy  
[AWS Elastic Beanstalk](#)  
Succeeded - 11 days ago  
[View logs](#)

[s32c45a](#) Source: requirements.txt