National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Sameera Bano
Student ID: 23244950

School of Computing
National College of Ireland

Supervisor: Shaguna Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sameera Bano
Student ID:	23244950
Programme:	Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Shaguna Gupta
Submission Due Date:	29/01/2025
Project Title:	Configuration Manual
Word Count:	836
Page Count:	14

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:	Sameera Bano
Date:	29th January 2025

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

Sameera Bano
23244950

Building a Flask Web Application with Machine Learning Models Using Docker

This guide details the steps to build and deploy a Flask web application that incorporates three machine learning models: Random Forest, Support Vector Machine (SVM), and Logistic Regression. The project is powered by Python and Docker for efficient development and deployment.

1 System Requirements

1.1 Hardware Requirements

- Operating System: Windows 10
- Processor: AMD Ryzen 3
- System Type: 64-bit Operating System
- RAM: 4 GB
- Storage:
  - SSD: 256 GB
  - HDD: 1 TB
- Display: HD with a refresh rate of 60 Hz

1.2 Software Requirements

- Python: Version 3.12
- VSCode: Version 1.95.3
- VSCode Extensions:
  - Pylance: Enabled
  - Docker Extension: Enabled
- Docker Desktop: Version 4.3.1

2. Installation Steps

1. Download the Flask Web App Code:
   Download the ZIP file containing the Flask application code and Docker configuration.

2. Extract the Files:
   Extract the ZIP file to a folder on your computer.

3. Open the Project in VSCode:
   Use VSCode to navigate to the extracted directory.

4. Run the Flask Application Locally:
   Start the application by running:
   ```bash
   python app.py
   ```

5. Install Dependencies:
   Install the required libraries, including Flask and machine learning packages, by running:
   ```bash
   pip install -r requirements.txt
   ```

3. Configuration

The application is designed to run in a containerized environment. The Dockerfile uses python:3.12-slim to minimize the container size while supporting all necessary libraries. Additionally, Docker Desktop was configured with WSL2 to ensure compatibility with Linux-based images.

4. Usage Instructions

4.1 Running the Flask Application

1. Start the Application:
   Run the app locally:
   ```bash
   python app.py
   ```
   The application will be available at http://localhost:5000.

2. Machine Learning Models Integration:
   The app includes the following models:
   - Random Forest: Predicts outcomes using an ensemble of decision trees.

- SVM: Performs classification tasks using hyperplanes.
- Logistic Regression: for checking if deployments were successful or not

3. Interact with the Application:
Use the web interface to input data and receive predictions from the models.

4.2 Deploying the Application with Docker

1. Create a Dockerfile:
```Dockerfile
# Use the official Python image
FROM python:3.12

# Set the working directory
WORKDIR /app

# Copy the dependencies file and install libraries
COPY requirements.txt /app/
RUN pip install --no-cache-dir -r requirements.txt

# Copy the Flask app and models
COPY . /app/

# Expose the port Flask will run on
EXPOSE 5000

# Run the Flask app
CMD ["python", "app.py"]
```

2. Build the Docker Image:
```bash
docker build -t flask-ml-app .
```

3. Run the Docker Container:
```bash
docker run -p 5000:5000 flask-ml-app
```

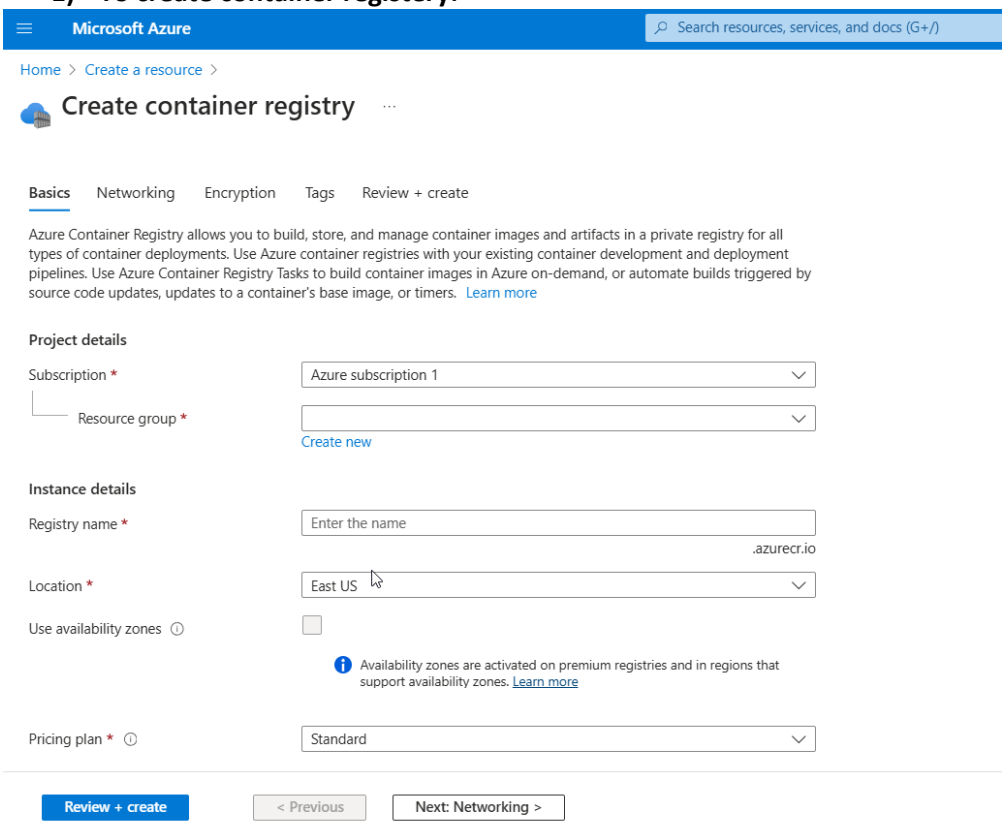
Key Features of the Application
- Random Forest: Predictive analytics using an ensemble learning method.
- SVM: Efficient classification of datasets with clear margins.

- Logistic Regression: for checked if deployments were successful or not.

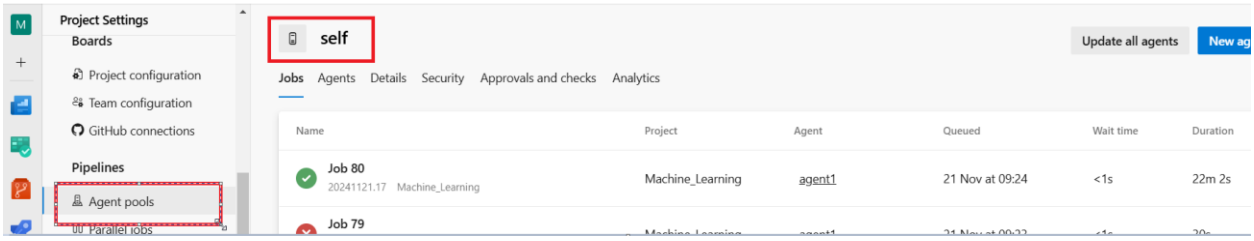
By combining these models, the application serves as a versatile tool for making predictions based on user inputs.

Screenshots:

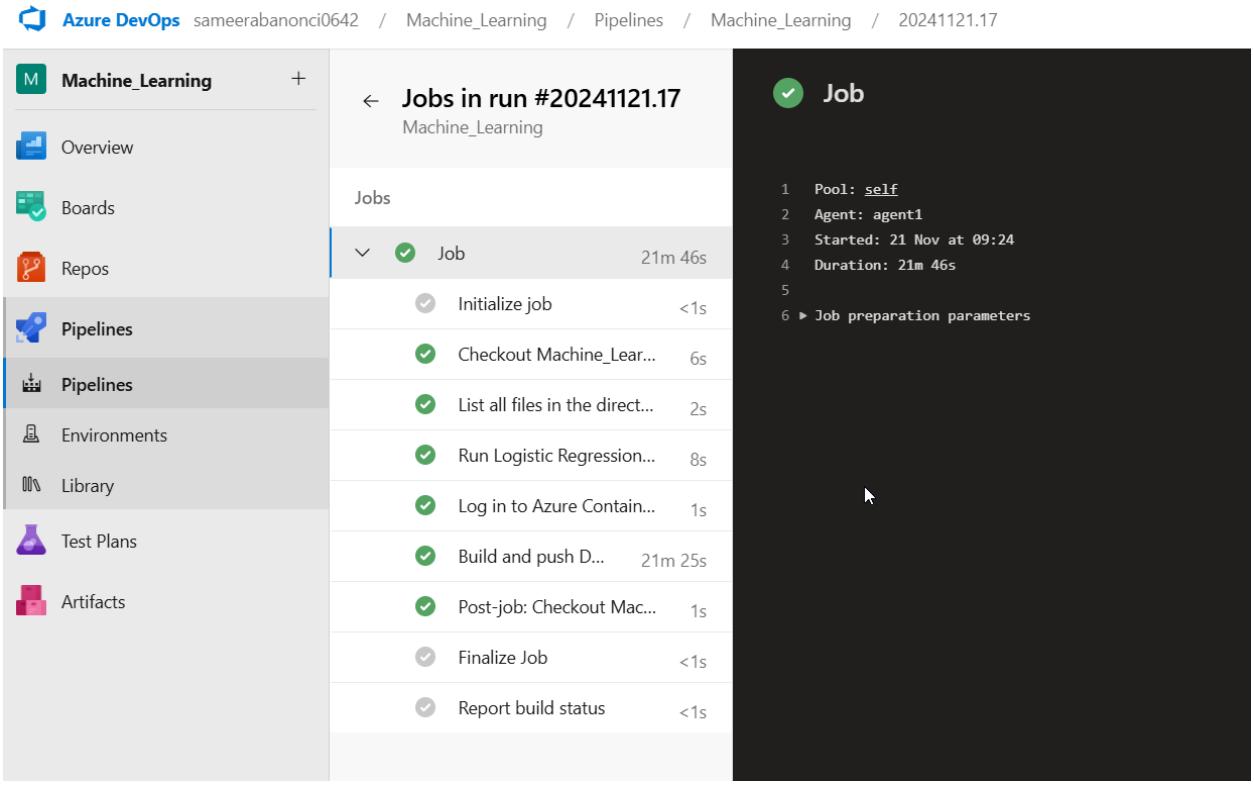
1) **To create container registery:**

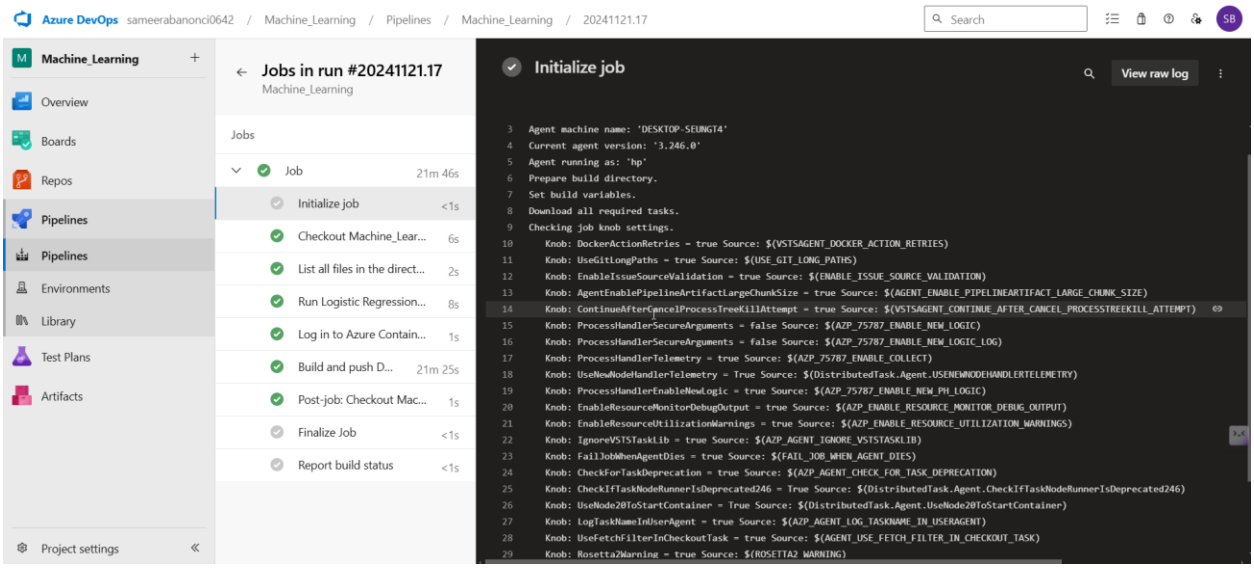


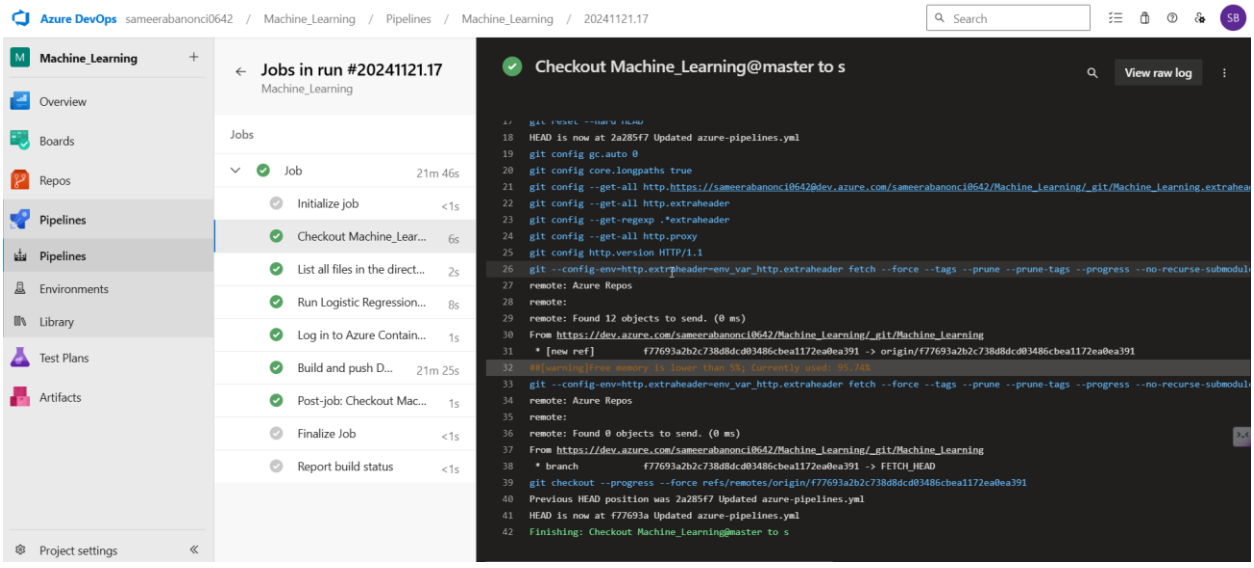
2) To run the pipeline I have used Agent pool with the name ‘self’

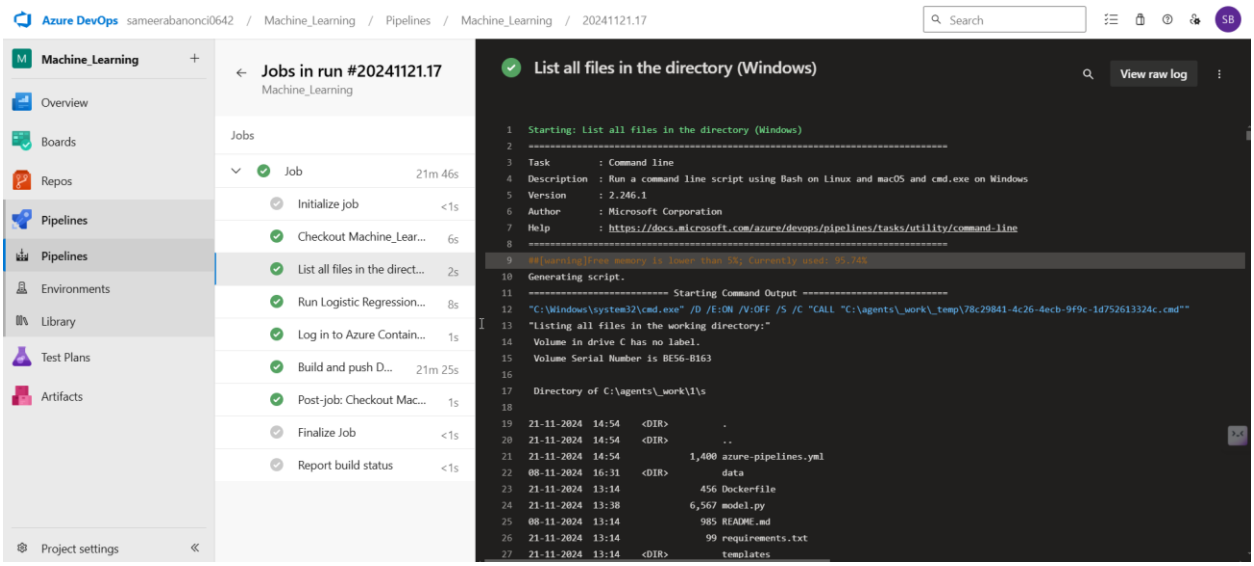


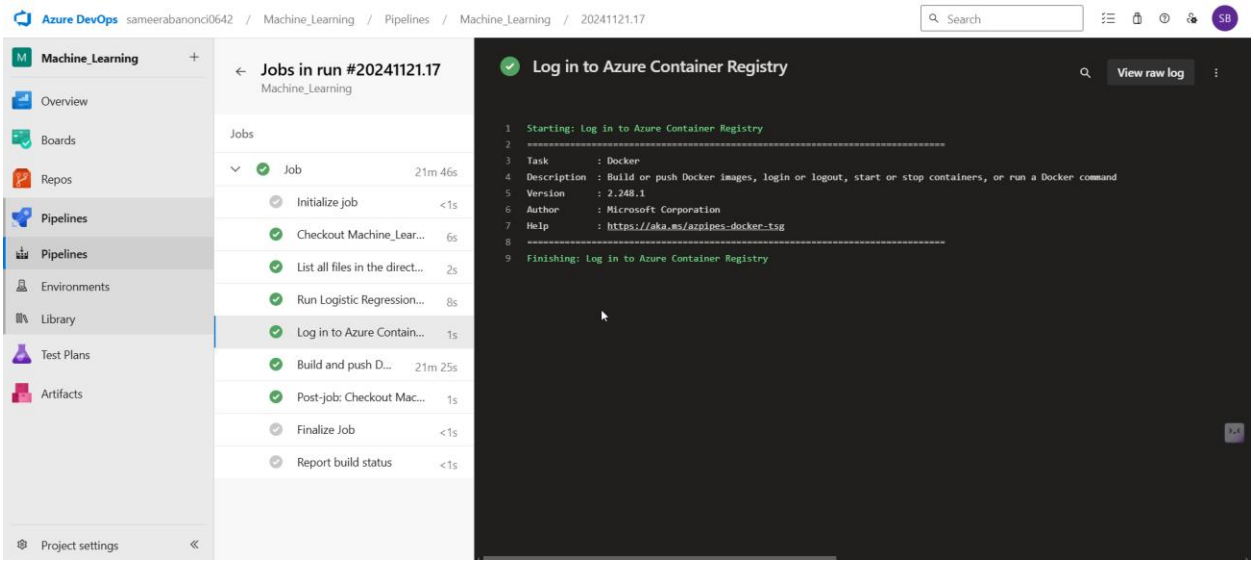
3) Jobs run in the pipeline :



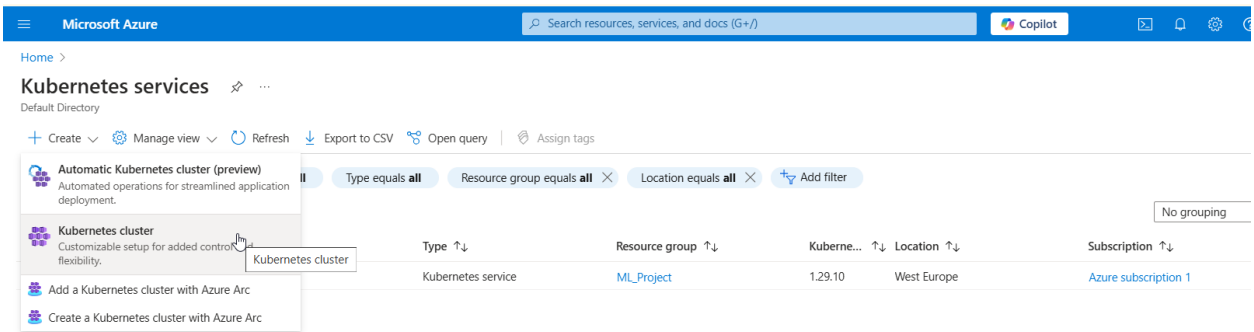

Azure DevOps sameerabanonci0642 / Machine_Learning / Pipelines / Machine_Learning / 20241121.17
Search
SB
Machine_Learning
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Library
Test Plans
Artifacts
Project settings
Jobs in run #20241121.17
Machine_Learning
Jobs
Job 21m 46s
Initialize job <1s
Checkout Machine_Lear... 6s
List all files in the direct... 2s
Run Logistic Regression... 8s
Log in to Azure Contain... 1s
Build and push D... 21m 25s
Post-job: Checkout Mac... 1s
Finalize Job <1s
Report build status <1s
Initialize job
View raw log
Agent machine name: 'DESKTOP-SEUNGT4'
Current agent version: '3.246.0'
Agent running as: 'hp'
Prepare build directory.
Set build variables.
Download all required tasks.
Checking job knob settings.
Knob: DockerActionRetries = true Source: $(VSTSAGENT_DOCKER_ACTION_RETRIES)
Knob: UseGitLongPaths = true Source: $(USE_GIT_LONG_PATHS)
Knob: EnableIssueSourceValidation = true Source: $(ENABLE_ISSUE_SOURCE_VALIDATION)
Knob: AgentEnablePipelineArtifactLargeChunkSize = true Source: $(AGENT_ENABLE_PIPELINEARTIFACT_LARGE_CHUNK_SIZE)
Knob: ContinueAfterCancelProcessTreeKillAttempt = true Source: $(VSTSAGENT_CONTINUE_AFTER_CANCEL_PROCESSTREEKILL_ATTEMPT)
Knob: ProcessHandlerSecureArguments = false Source: $(AZP_75787_ENABLE_NEW_LOGIC)
Knob: ProcessHandlerSecureArguments = false Source: $(AZP_75787_ENABLE_NEW_LOGIC_LOG)
Knob: ProcessHandlerTelemetry = true Source: $(AZP_75787_ENABLE_COLLECT)
Knob: UseNewNodeHandlerTelemetry = True Source: $(DistributedTask.Agent.USENEWNODEHANDLERTELEMETRY)
Knob: ProcessHandlerEnableNewLogic = true Source: $(AZP_75787_ENABLE_NEW_PH_LOGIC)
Knob: EnableResourceMonitorDebugOutput = true Source: $(AZP_ENABLE_RESOURCE_MONITOR_DEBUG_OUTPUT)
Knob: EnableResourceUtilizationWarnings = true Source: $(AZP_ENABLE_RESOURCE_UTILIZATION_WARNINGS)
Knob: IgnoreVSTSTaskLib = true Source: $(AZP_AGENT_IGNORE_VSTSTASKLIB)
Knob: FailJobWhenAgentDies = true Source: $(FAIL_JOB_WHEN_AGENT_DIES)
Knob: CheckForTaskDeprecation = true Source: $(AZP_AGENT_CHECK_FOR_TASK_DEPRECATION)
Knob: CheckIfTaskNodeRunnerIsDeprecated246 = True Source: $(DistributedTask.Agent.CheckIfTaskNodeRunnerIsDeprecated246)
Knob: UseNode20ToStartContainer = True Source: $(DistributedTask.Agent.UseNode20ToStartContainer)
Knob: LogTaskNameInUserAgent = true Source: $(AZP_AGENT_LOG_TASKNAME_IN_USERAGENT)
Knob: UseFetchFilterInCheckoutTask = true Source: $(AGENT_USE_FETCH_FILTER_IN_CHECKOUT_TASK)
Knob: Rosetta2Warning = true Source: $(ROSETTA2_WARNING)


Azure DevOps sameerabanonci0642 / Machine_Learning / Pipelines / Machine_Learning / 20241121.17
Search
SB
Machine_Learning
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Library
Test Plans
Artifacts
Project settings
Jobs in run #20241121.17
Machine_Learning
Jobs
Job 21m 46s
Initialize job <1s
Checkout Machine_Lear... 6s
List all files in the direct... 2s
Run Logistic Regression... 8s
Log in to Azure Contain... 1s
Build and push D... 21m 25s
Post-job: Checkout Mac... 1s
Finalize Job <1s
Report build status <1s
Checkout Machine_Learning@master to s
View raw log
HEAD is now at 2a285f7 Updated azure-pipelines.yml
git config gc.auto 0
git config core.longpaths true
git config --get-all http.https://sameerabanonci0642@dev.azure.com/sameerabanonci0642/Machine_Learning/_git/Machine_Learning.extraheader
git config --get-all http.extraheader
git config --get-regexp .*extraheader
git config --get-all http.proxy
git config http.version HTTP/1.1
git --config-env=http.extraheader=env_var_http.extraheader fetch --force --tags --prune --prune-tags --progress --no-recurse-submodules
remote: Azure Repos
remote:
remote: Found 12 objects to send. (0 ms)
From https://dev.azure.com/sameerabanonci0642/Machine_Learning/_git/Machine_Learning
* [new ref] f77693a2b2c738d8dcd03486cbea1172ea0ea391 -> origin/f77693a2b2c738d8dcd03486cbea1172ea0ea391
##[warning]Free memory is lower than 5%, Currently used: 95.74%
git --config-env=http.extraheader=env_var_http.extraheader fetch --force --tags --prune --prune-tags --progress --no-recurse-submodules
remote: Azure Repos
remote:
remote: Found 0 objects to send. (0 ms)
From https://dev.azure.com/sameerabanonci0642/Machine_Learning/_git/Machine_Learning
* branch f77693a2b2c738d8dcd03486cbea1172ea0ea391 -> FETCH_HEAD
git checkout --progress --force refs/remotes/origin/f77693a2b2c738d8dcd03486cbea1172ea0ea391
Previous HEAD position was 2a285f7 Updated azure-pipelines.yml
HEAD is now at f77693a Updated azure-pipelines.yml
Finishing: Checkout Machine_Learning@master to s

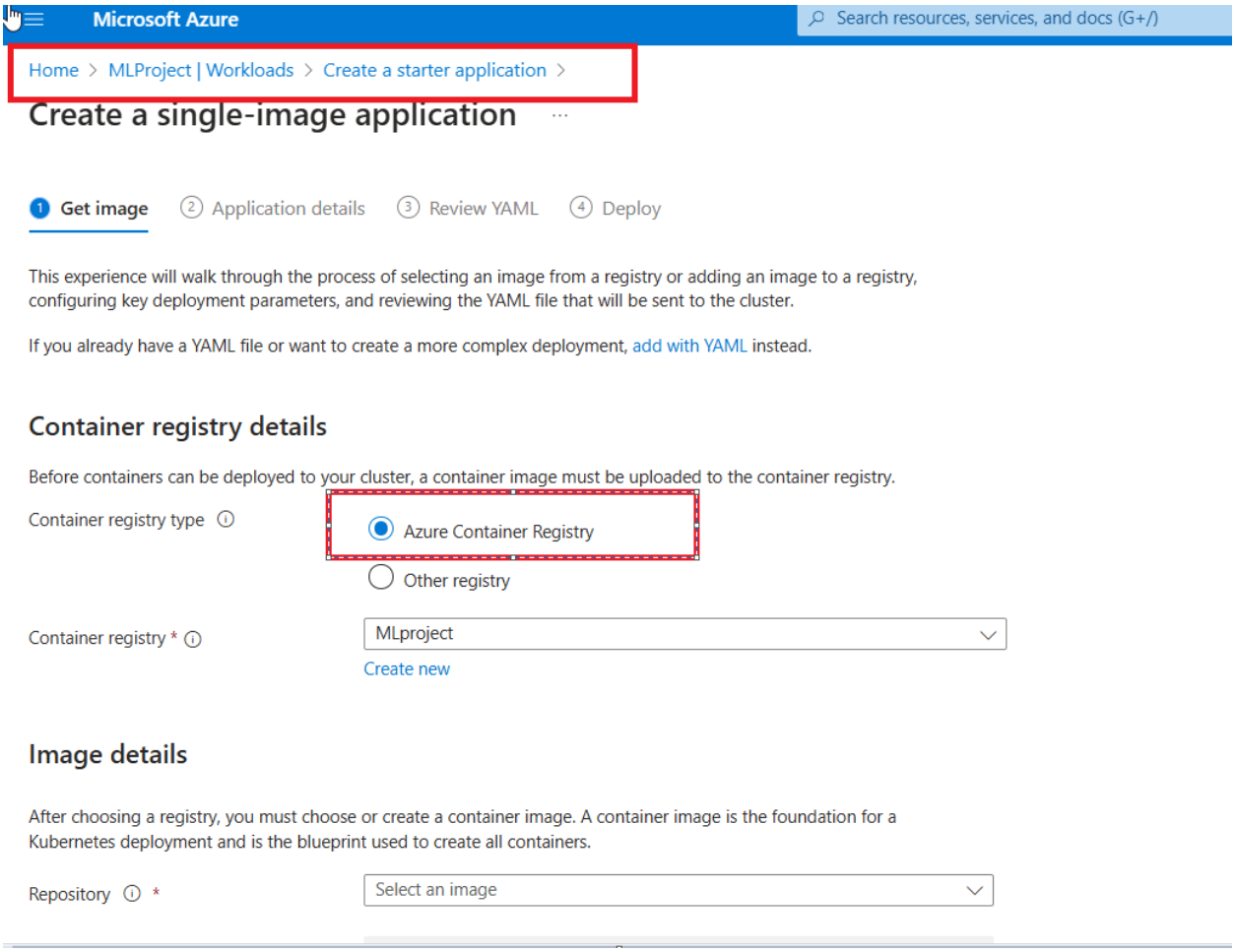




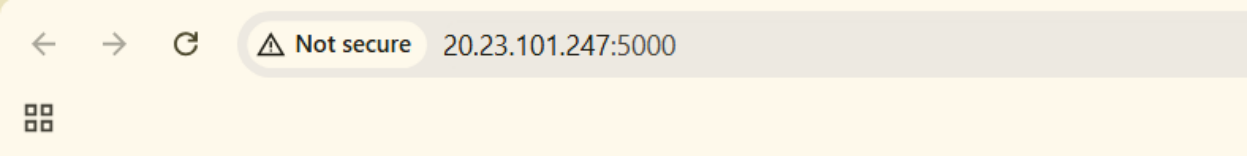
➢ To create the Kubernetes cluster:



➢ To create a starter application to be deployed :



- Public Address to check Machine Learning models evaluation in Azure- http://20.23.101.247:5000/



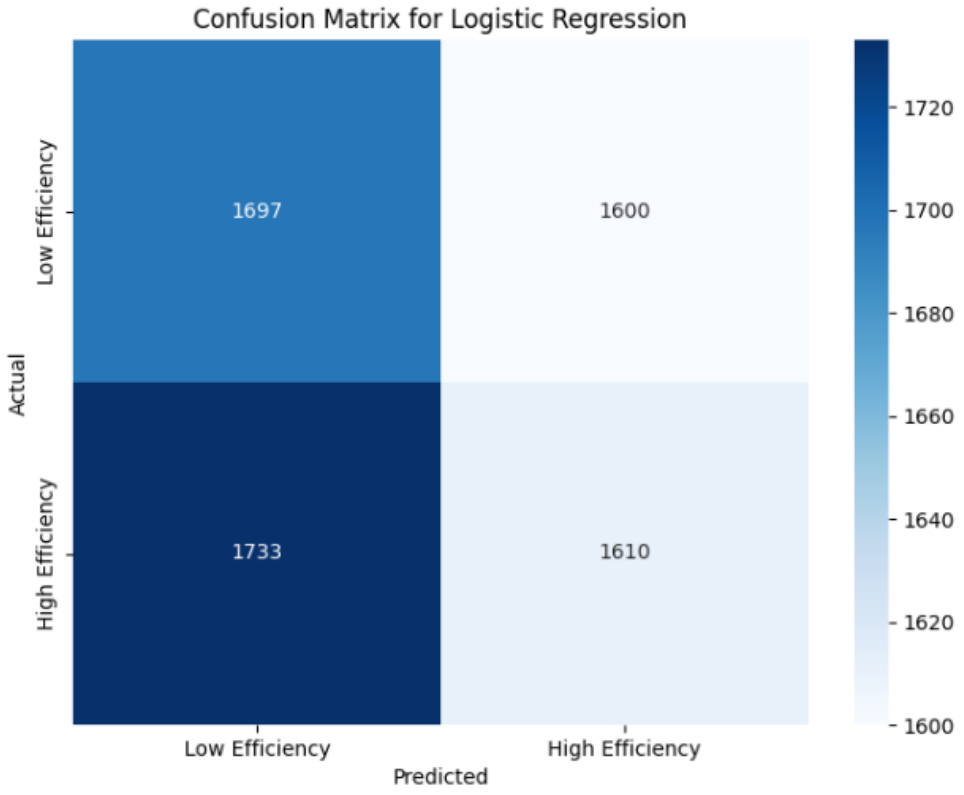
Model Evaluation Results

Logistic Regression

Accuracy: 0.4980421686746988

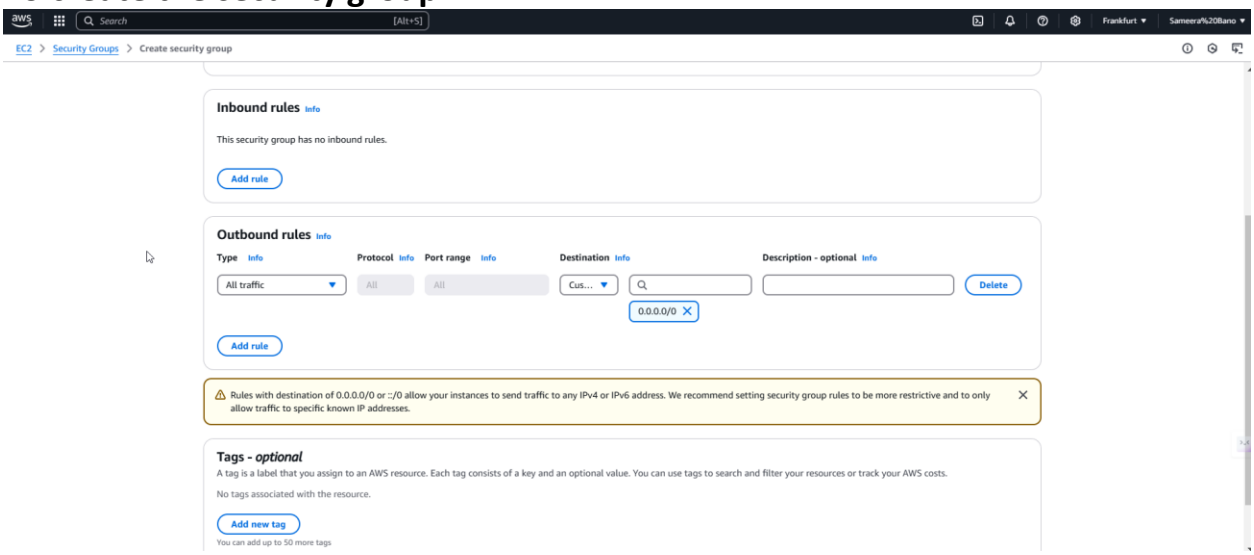
	precision	recall	f1-score	support
0	0.49	0.51	0.50	3297
1	0.50	0.48	0.49	3343
accuracy			0.50	6640
macro avg	0.50	0.50	0.50	6640
weighted avg	0.50	0.50	0.50	6640

Confusion Matrix:

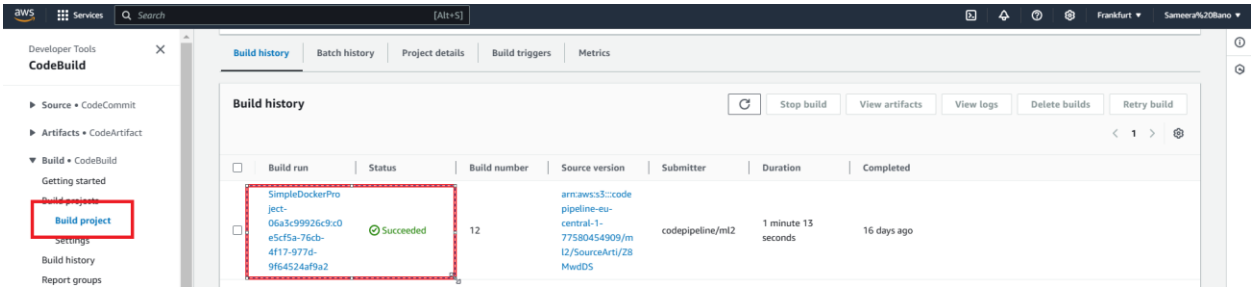


AWS:

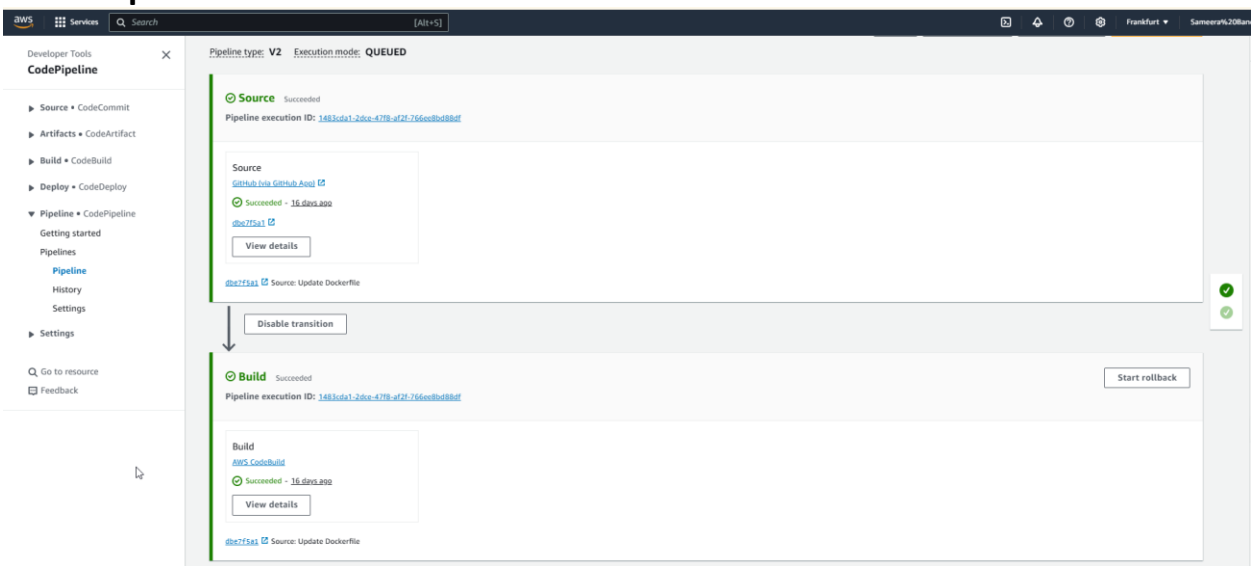
- **To create the security group:**



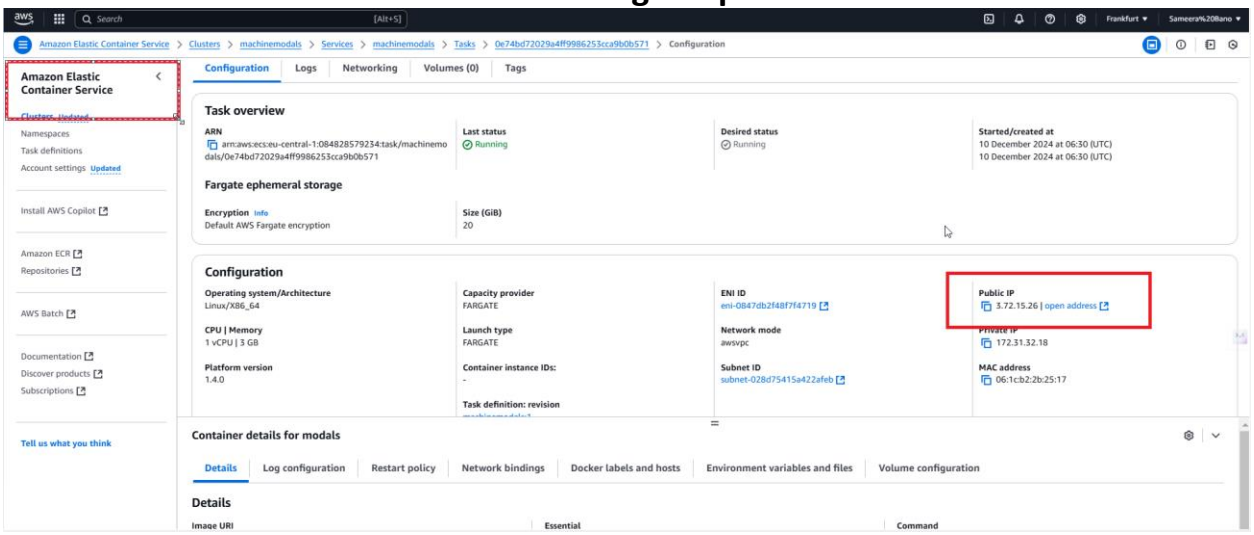
- **CodeBuild:**



➢ Pipeline:



➢ Elastic Container Service containing the public IP:



AWS public IP to analyse Machine learning models evaluation- http://3.72.15.26:5000/

Evaluation result:

Model Evaluation Results

Logistic Regression

Accuracy: 0.4935

	precision	recall	f1-score	support
0	0.51	0.38	0.43	5117
1	0.49	0.61	0.54	4883
accuracy			0.49	10000
macro avg	0.50	0.50	0.49	10000
weighted avg	0.50	0.49	0.49	10000

Confusion Matrix: