Configuration Manual

MSc Research Project
Cloud Computing

Vinay Kalidindi
Student ID: 23107316

School of Computing
National College of Ireland

Supervisor: Shivani Jaswal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Vinay Kalidindi
Student ID:	23107316
Programme:	Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Shivani Jaswal
Submission Due Date:	29/01/2025
Project Title:	Configuration Manual
Word Count:	693
Page Count:	8

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:	Vinay Kalidindi
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

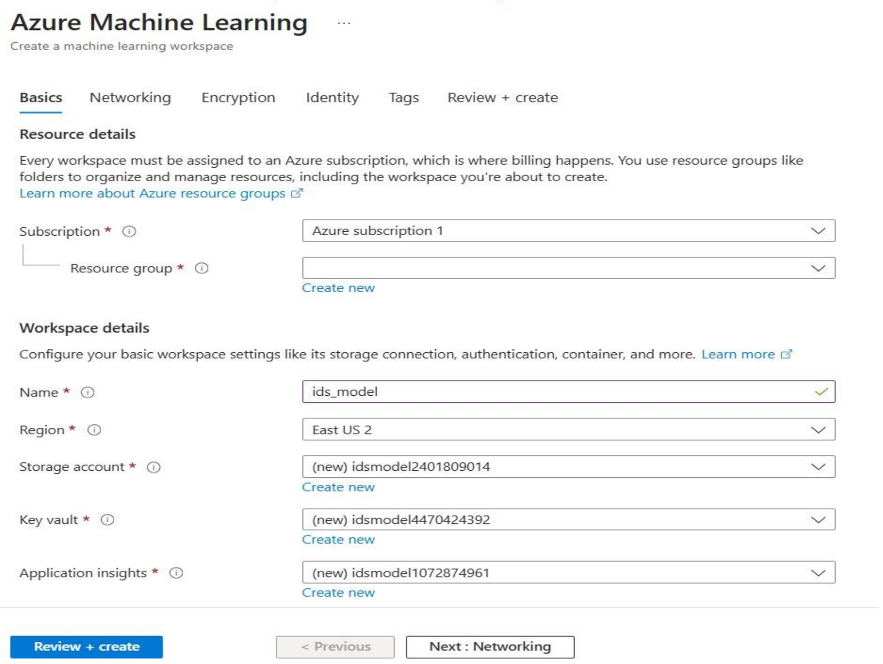
Vinay Kalidindi

x 23107316

1. Creating Azure Resource group and Workspace

To create the workspace:

- Log in to the Azure Portal.
- Navigate to Azure Machine Learning.
- Create a new Resource Group and Workspace as shown in Figure 1.



Azure Machine Learning

Create a machine learning workspace

Resource details

Every workspace must be assigned to an Azure subscription, which is where billing happens. You use resource groups like folders to organize and manage resources, including the workspace you're about to create.
Learn more about Azure resource groups

Subscription * : Azure subscription 1

Resource group * :

Workspace details

Configure your basic workspace settings like its storage connection, authentication, container, and more. Learn more

Name * : ids_model

Region * : East US 2

Storage account * : (new) idsmodel2401809014

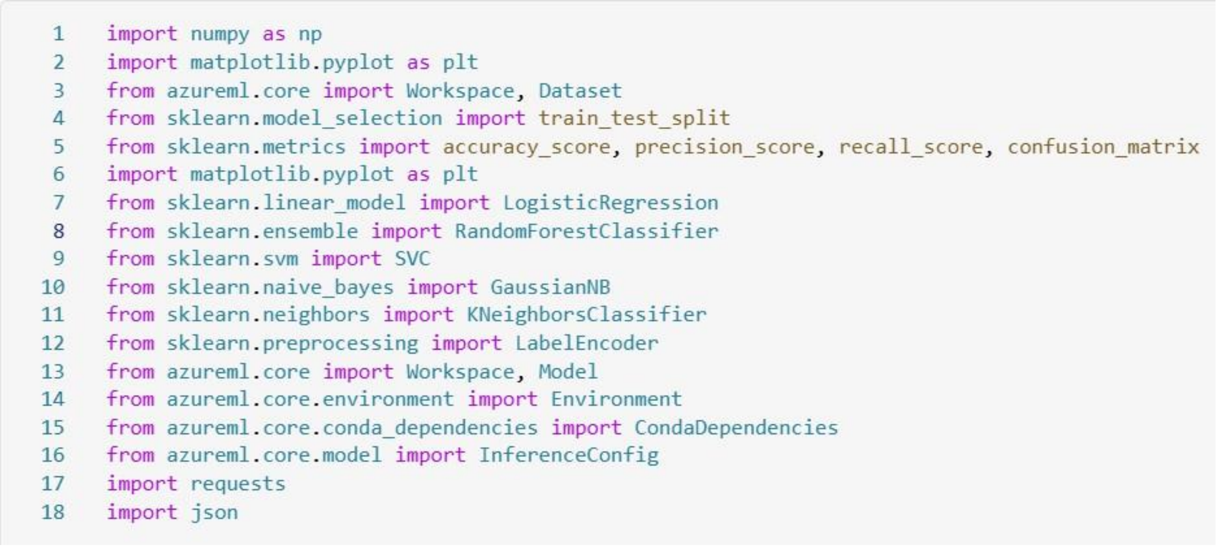
Key vault * : (new) idsmodel4470424392

Application insights * : (new) idsmodel1072874961

Figure 1: Azure ML Workspace Creation

2. Importing Libraries

Libraries required for data preprocessing, model training, and deployment were imported into Azure Notebooks. This setup is shown in Figure 2.

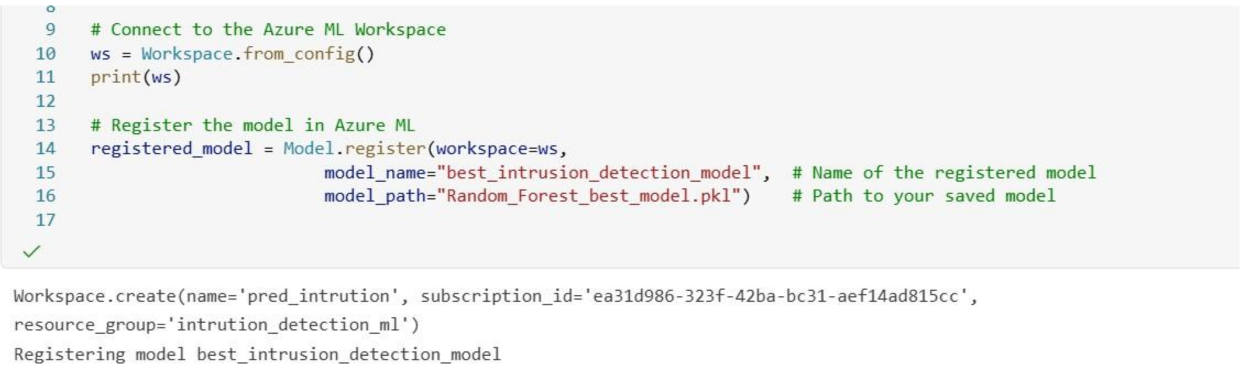


```python
import numpy as np
import matplotlib.pyplot as plt
from azureml.core import Workspace, Dataset
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import LabelEncoder
from azureml.core import Workspace, Model
from azureml.core.environment import Environment
from azureml.core.conda_dependencies import CondaDependencies
from azureml.core.model import InferenceConfig
import requests
import json
```

Figure 2: Library Imports

3. ML Model Registration

The trained Random Forest model (Random_Forest_best_model.pkl) was registered in the Azure ML Workspace. The Python script for model registration is as shown in Figure 3.



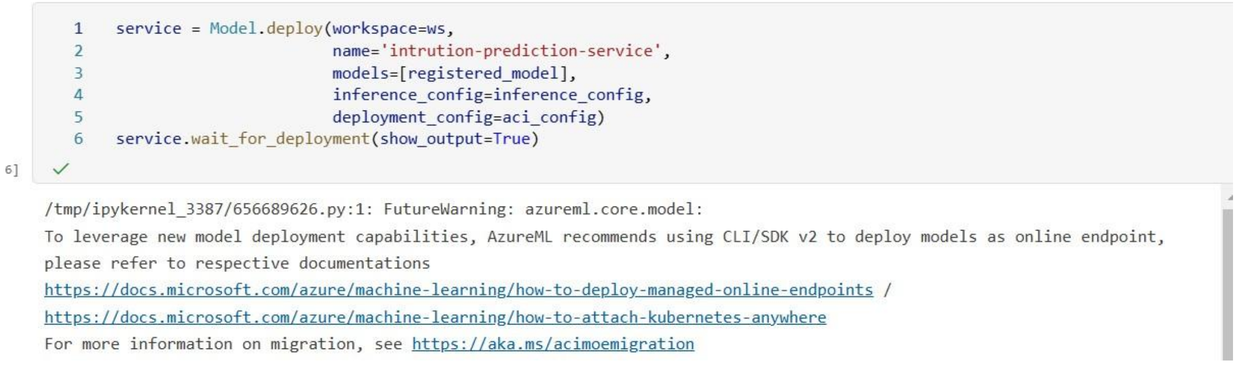
```python
# Connect to the Azure ML Workspace
ws = Workspace.from_config()
print(ws)

# Register the model in Azure ML
registered_model = Model.register(workspace=ws,
                       model_name="best_intrusion_detection_model",  # Name of the registered model
                       model_path="Random_Forest_best_model.pkl")    # Path to your saved model

```

```
Workspace.create(name='pred_intrution', subscription_id='ea31d986-323f-42ba-bc31-aef14ad815cc',
resource_group='intrution_detection_ml')
Registering model best_intrusion_detection_model
```

Figure 3: ML Model Registration

4. Deploying Model

The registered model was deployed as a Real-Time Endpoint. Deployment settings and the deployment script are as in Figure 4



```
service = Model.deploy(workspace=ws,
                       name='intrution-prediction-service',
                       models=[registered_model],
                       inference_config=inference_config,
                       deployment_config=aci_config)
service.wait_for_deployment(show_output=True)
```

```
/tmp/ipykernel_3387/656689626.py:1: FutureWarning: azureml.core.model:
To leverage new model deployment capabilities, AzureML recommends using CLI/SDK v2 to deploy models as online endpoint,
please refer to respective documentations
https://docs.microsoft.com/azure/machine-learning/how-to-deploy-managed-online-endpoints /
https://docs.microsoft.com/azure/machine-learning/how-to-attach-kubernetes-anywhere
For more information on migration, see https://aka.ms/acimoemigration
```

Figure 4: Model Deployment

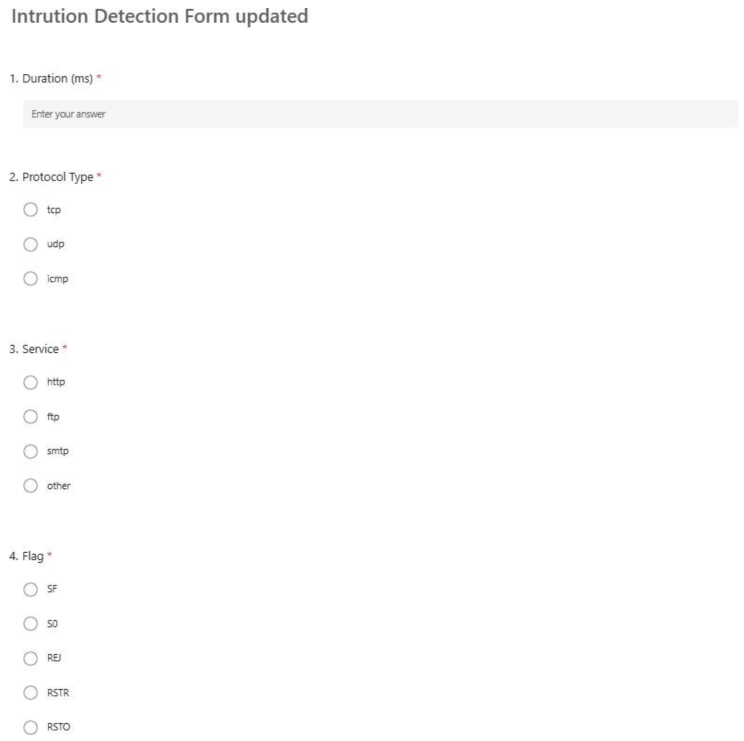
5. Microsoft Forms Setup

Step 1: Form Creation

- Open **Microsoft Forms** and create a new form titled "Intrusion Detection Data Entry".
- Add fields for key attributes:
    - **Duration (ms)** (Number)
    - **Source Bytes** (Number)
    - **Destination Bytes** (Number)
    - **Protocol Type** (Dropdown: TCP, UDP, ICMP)
    - **Service** (Dropdown: HTTP, FTP, SMTP, Other)
    - **Flag** (Dropdown: SF, S0, REJ, etc.)
    - **Wrong Fragment** (Dropdown: 0, 1, 2)

Step 2: Set Required Fields

- Make all fields required to avoid missing data.



Intrution Detection Form updated

1. Duration (ms) *

Enter your answer

2. Protocol Type *

○ tcp

○ udp

○ icmp

3. Service *

○ http

○ ftp

○ smtp

○ other

4. Flag *

○ SF

○ S0

○ REJ

○ RSTR

○ RSTO

Figure 5: Forms Setup

6. Configuring Power Automate

Step 1: Trigger Setup

- Trigger: **"When a new response is submitted"** (Microsoft Forms).
- Choose the form created in the previous step.

Step 2: Fetch Response Details

- Action: **"Get response details".**
- Link it to the form and fetch all responses dynamically.

Step 3: Data Transformation

- Initialize variables for each field:
  - Example: protocol_type, service, flag as Strings.
  - Use **Switch Actions** to encode categorical values (e.g., TCP = 0, HTTP = 1).

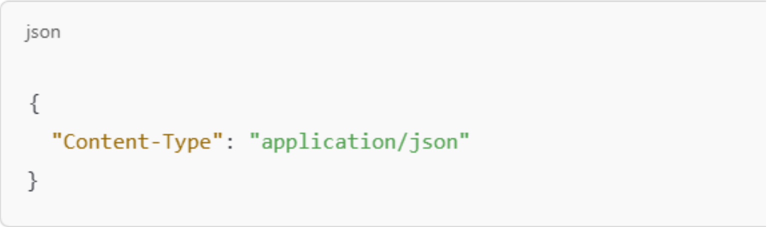
Step 4: Compose Input Data

- Use the **Compose** action to format the data into a list: [

```
@{variables('duration')},
@{variables('protocol_type_encoded')},
@{variables('service_encoded')},
@{variables('flag_encoded')},
 @{variables('src_bytes')},  @{variables('dst_bytes')},
 0, @{variables('wrong_fragment')}, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 3,
 0, 0, 0, 0, 1, 0, 0, 100, 100, 0, 0, 0, 0, 0, 0, 0, 0
]
```

Step 5: HTTP Request to ML Model

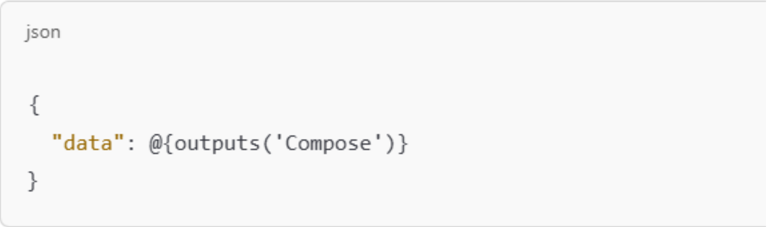
- Action: **"HTTP"**
  - **Method:** POST
  - **URI:** Scoring URI from Azure ML deployment.



json

```json
{
  "Content-Type": "application/json"
}
```

- Body:



json

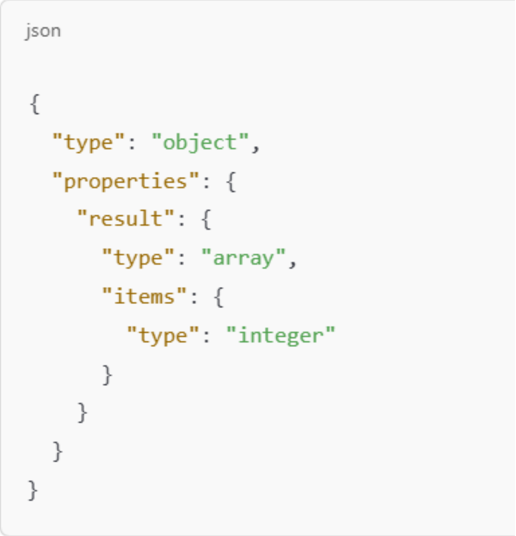
```json
{
  "data": @{outputs('Compose')}
}
```

○

Figure 7: HTTP Request Header and Body

Step 6: Parse JSON

- Parse the HTTP response to extract the ML model’s prediction:



json

```json
{
  "type": "object",
  "properties": {
    "result": {
      "type": "array",
      "items": {
        "type": "integer"
      }
    }
  }
}
```

Figure 8: Json parse for Model prediction

Step 7: Conditional Check

- Action: **Condition**
  - **Condition:** result [0] is equal to 1
    - **If Yes:** Send an alert email (e.g., "Intrusion Detected").
    - **If No:** Log the event as normal traffic.