



AI on IBM Z

Fraud detection solution template

This solution template provides an example on how to deploy AI using a Linux on Z environment, while making use of open source frameworks, Triton Inference Server (TIS), and more.

Within this solution template, there are various phases of the AI lifecycle included. Work through each of the following steps to deploy your own fraud detection solution on IBM Z.

Table of contents

AI model training.....	3
AI model deployment.....	8
AI model analysis.....	12
AI model integration.....	17



Step 1

AI model training

We will build a fraud detection AI model by training with the provided rapid AI on IBM Z development Jupyter notebook. Simply point the Jupyter notebook to your dataset and run it to generate your AI model. This trained AI model can then be deployed with MLz.

All sample code for this section is within

C:\Users\Administrator\Documents\ai-solution-templates

Environments used

Windows

Dataset guidance

Sample open source credit card transaction dataset can be found on Kaggle -

C:\Users\Administrator\Documents\ai-solution-templates\datasets\credit_card_transactions-ibm_v2.csv

Due to the size of the sample dataset, the provided Jupyter notebook takes a subset of the data to decrease the training time. Please modify the code in the “Fetch and process data” cell of the provided Jupyter notebook later to use more data during training.

It can also be found on Kaggle -

<https://www.kaggle.com/datasets/ealtman2019/credit-card-transactions>

Required features

- User (integer) – unique ID for user making transaction
- Card (integer) – unique ID for credit card
- Year (integer) – year of the transaction
- Month (integer) – month of the transaction
- Day (integer) – day of the month of the transaction
- Time (integer) - time of the transaction (HH:MM)
- Amount (float) – dollar amount of the transaction
- Use Chip (string) – the type of transaction
- Merchant Name (integer) – unique ID for merchant name
- Zip (integer) – zip code of the transaction

1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

[Access rapid AI on IBM Z development environment](#)

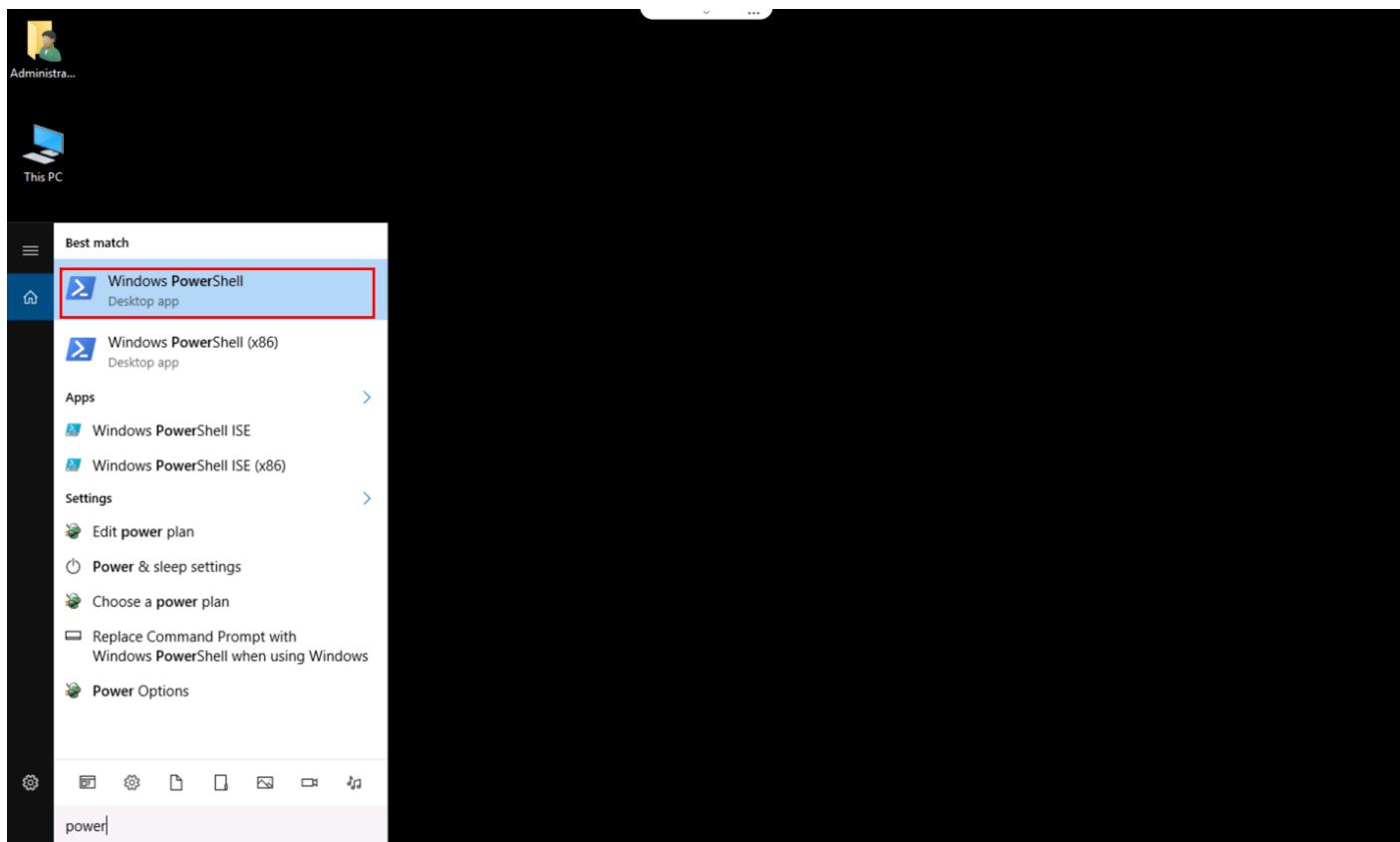
Provide data

Model training

Access trained AI model

Access rapid AI on IBM Z development environment

1. Open Windows PowerShell



1. AI model training.

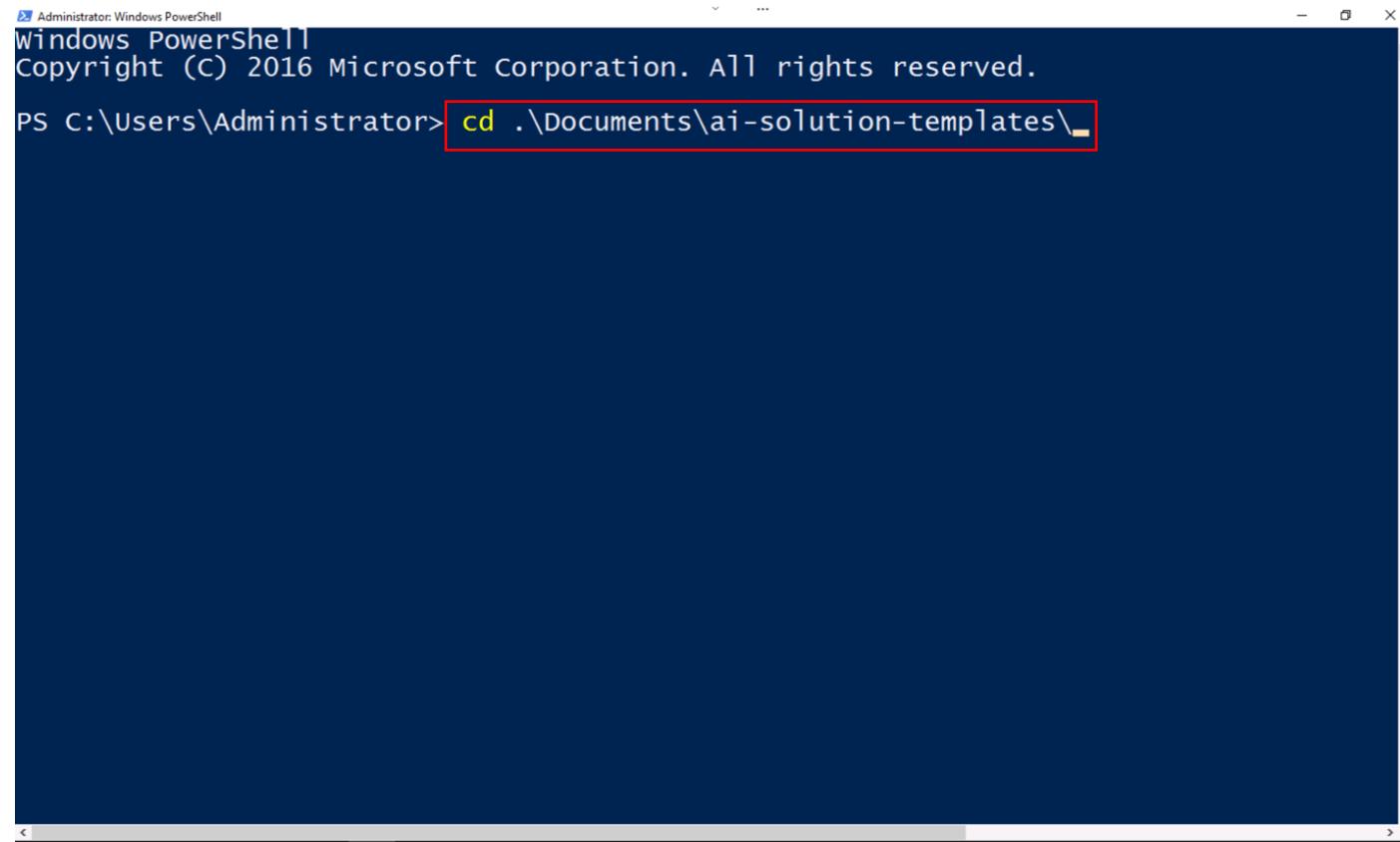
2. AI model deployment

3. AI model analysis

4. AI model integration

2. Go to model training pipeline

```
cd .\Documents\ai-solution-templates\
```



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window shows the following text:
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.
PS C:\Users\Administrator> **cd .\Documents\ai-solution-templates**

The command `cd .\Documents\ai-solution-templates\` is highlighted with a red rectangular box.

1. AI model training.

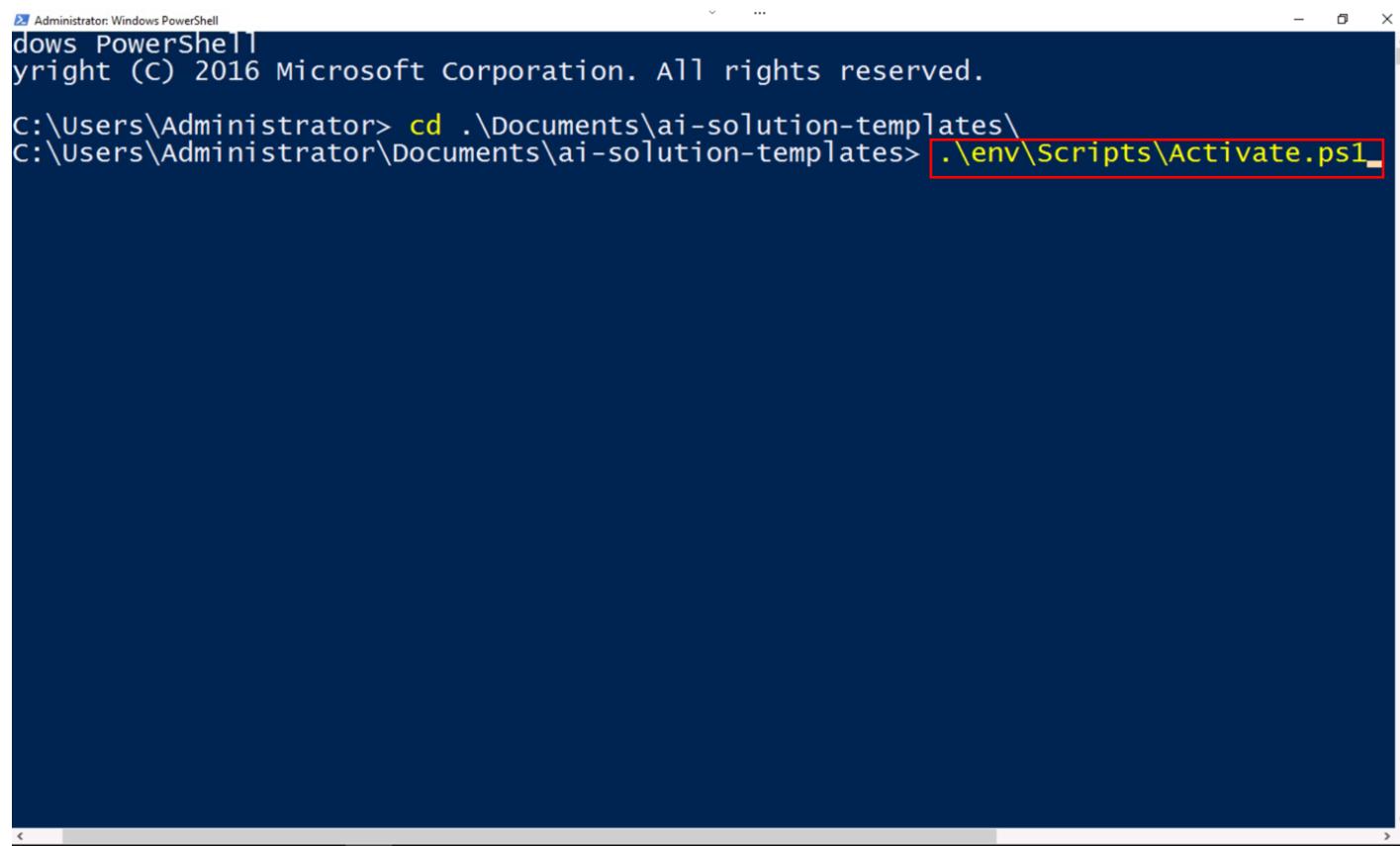
2. AI model deployment

3. AI model analysis

4. AI model integration

3. Activate Python virtual environment

`.\env\Scripts\Activate.ps1`



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window shows the following command being run:

```
Administrator: Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator> cd .\Documents\ai-solution-templates
C:\Users\Administrator\Documents\ai-solution-templates> .\env\Scripts\Activate.ps1
```

The path ".\env\Scripts\Activate.ps1" is highlighted with a red rectangle.

1. AI model training.

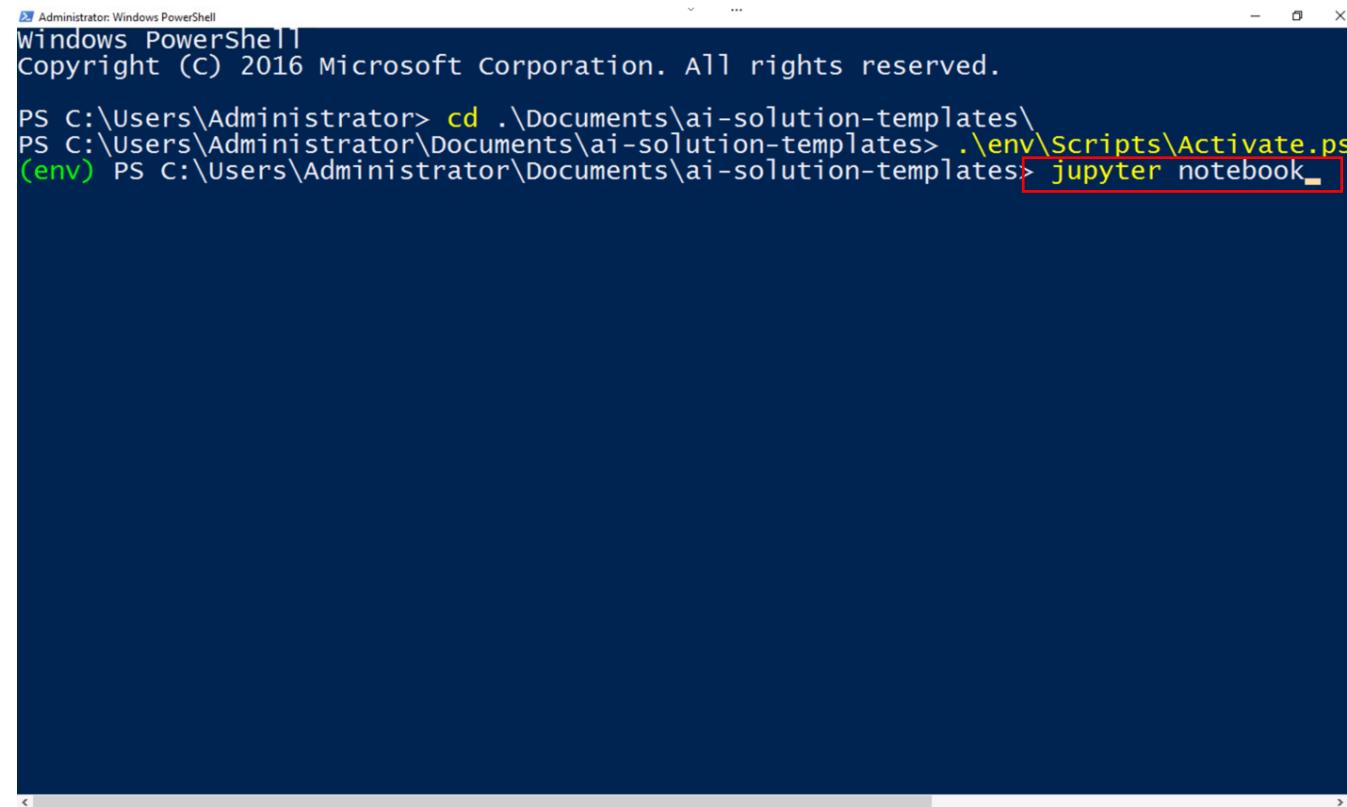
2. AI model deployment

3. AI model analysis

4. AI model integration

4. Run Jupyter

jupyter notebook



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window shows the following command being run:

```
PS C:\Users\Administrator> cd .\Documents\ai-solution-templates\  
PS C:\Users\Administrator\Documents\ai-solution-templates> .\env\scripts\Activate.ps1  
(env) PS C:\Users\Administrator\Documents\ai-solution-templates> jupyter notebook
```

The last line of the command, "jupyter notebook", is highlighted with a red rectangular box.

1. AI model training.

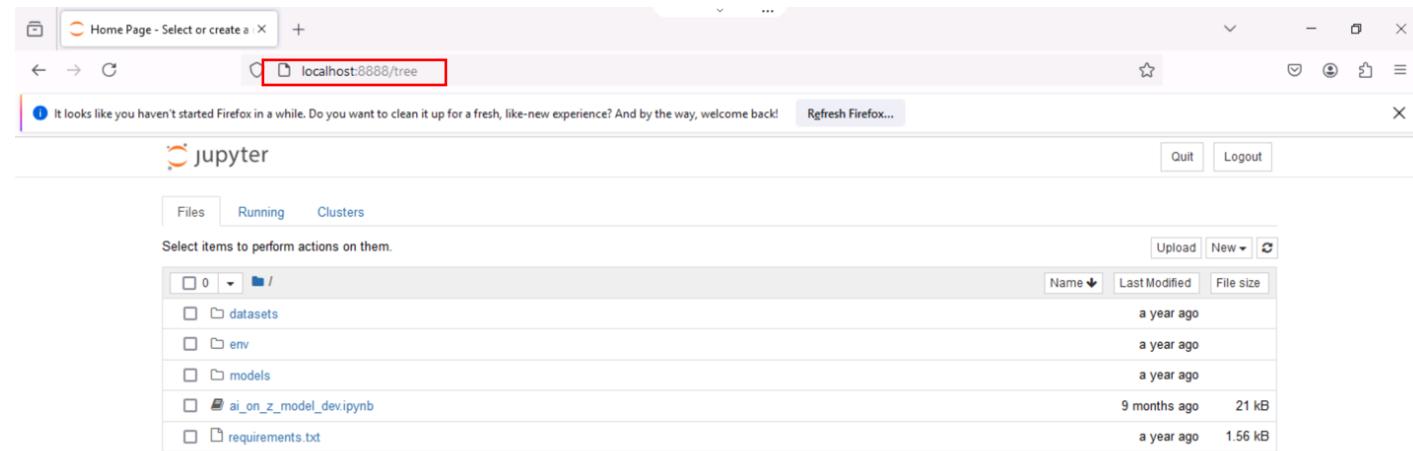
2. AI model deployment

3. AI model analysis

4. AI model integration

5. View Jupyter interface

- a. Go to localhost:8888 in a web browser



1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

Access rapid AI on IBM Z development environment

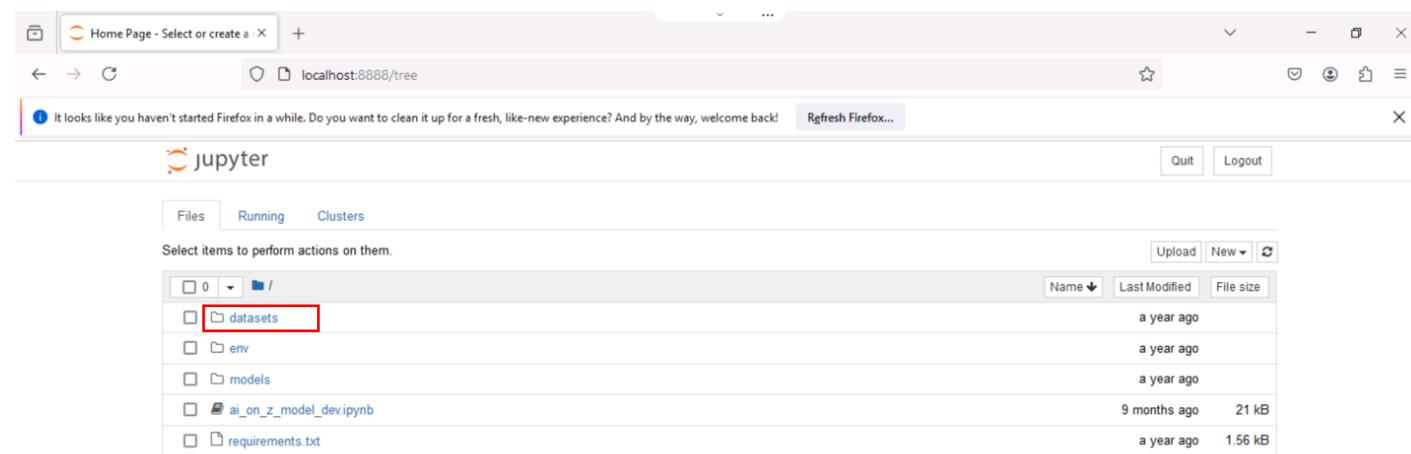
[Provide data](#)

[Model training](#)

Access trained AI model

Provide data (optional)

1. Your input dataset (csv) in datasets/ directory



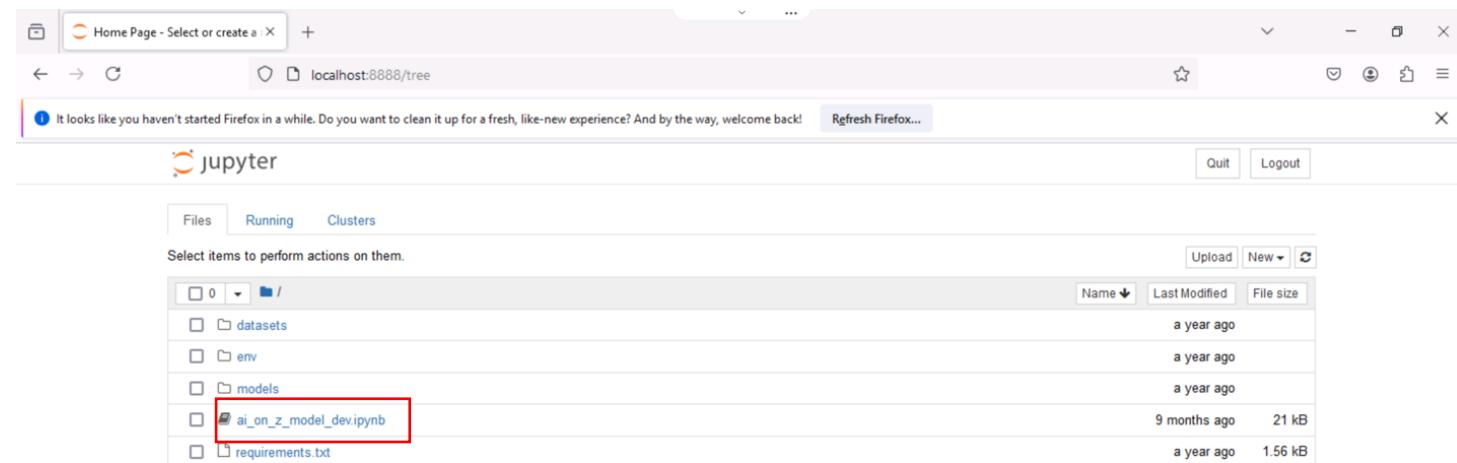
1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

2. Add input data to Jupyter notebook
(ai_on_z_model_dev.ipynb)



- a. Set **DATASET_FILENAME** to the path to your dataset
- b. Set **DATASET_LABEL_NAME** to the name of the column you're predicting from the dataset

The screenshot shows a Jupyter Notebook interface with the title "AI on IBM Z Model Development". The notebook contains three code cells:

- Import required python packages**:

```
In [ ]: import numpy as np
import pandas as pd
import json
import time

# Model training
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import GradientBoostingClassifier, RandomForestRegressor, Random Forest Regressor
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

# Data preprocessing
from sklearn.preprocessing import OrdinalEncoder, StandardScaler

# PMML
from sklearn2pmml import sklearn2pmml
from sklearn.compose import ColumnTransformer
from sklearn2pmml.pipeline import PMMLPipeline
```
- Input dataset and label**:

```
In [ ]: # User must provide filepath to dataset and label name

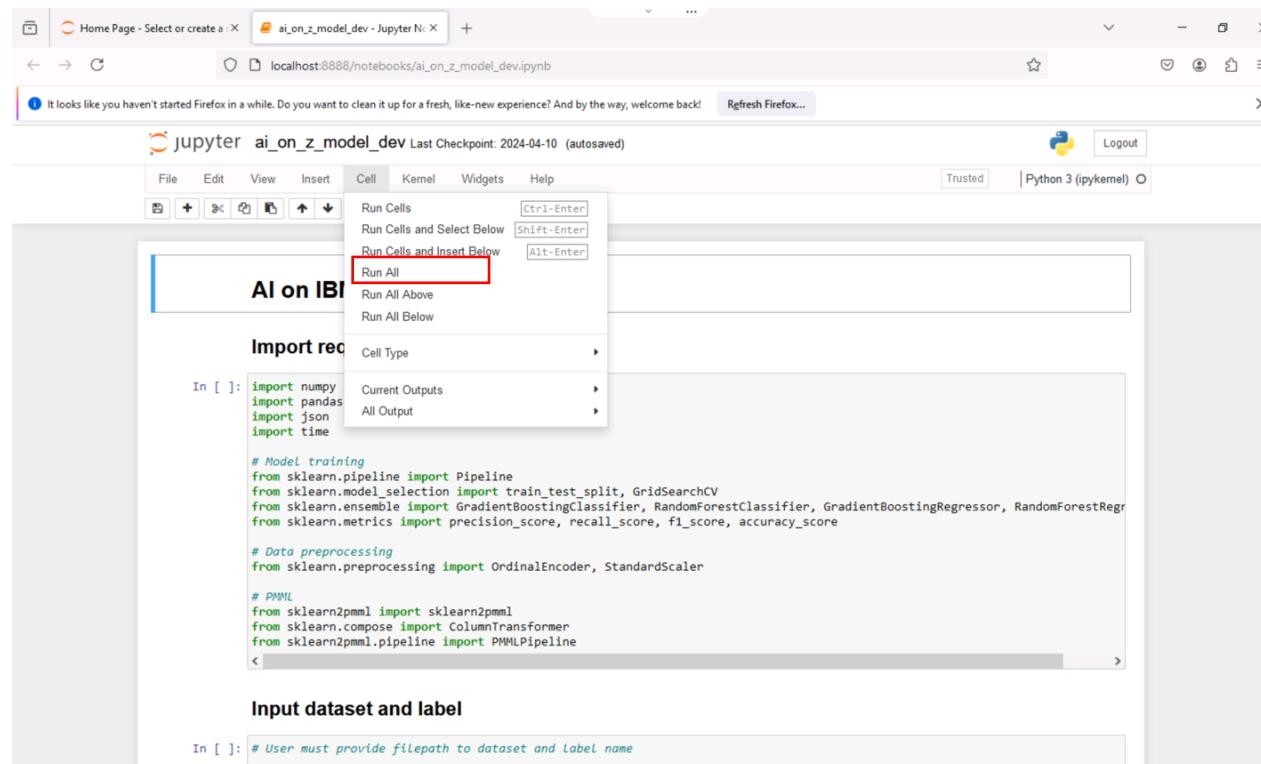
# Available datasets / label names:
# - credit_card_transactions-ibm_v2.csv / Is Fraud?
# - Data_Premium_V1.csv / insurance_cost
# - credit_risk_dataset.csv / loan_status
DATASET_FILENAME = 'datasets/credit_card_transactions-ibm_v2.csv'
DATASET_LABEL_NAME = 'Is Fraud?'
```
- Split features and labels from dataset**:

```
In [ ]: def split_features_and_labels(dataset_df, label):
    features = dataset_df.copy()
```

Model training

1. Step through and run all cells within Jupyter notebook (ai_on_z_model_dev.ipynb) within web browser

Note: This may take several minutes



The screenshot shows a Firefox browser window with a Jupyter Notebook interface. The title bar says 'jupyter ai_on_z_model_dev Last Checkpoint: 2024-04-10 (autosaved)'. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3 (ipykernel). A context menu is open over the first code cell, with the 'Run All' option highlighted by a red box. The code cell contains imports for numpy, pandas, json, time, and various sklearn modules. Below the code cell, there's a section titled 'Input dataset and label' with a placeholder 'In []: # User must provide filepath to dataset and Label name'.

Access rapid AI on IBM Z development environment

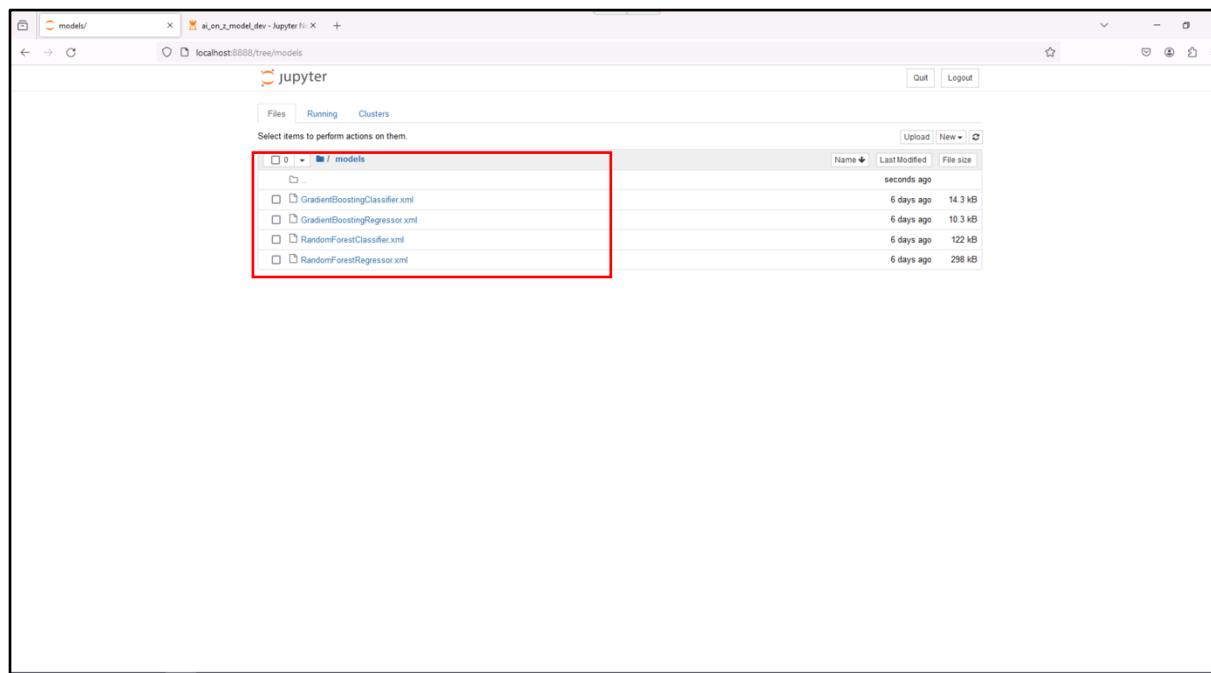
Provide data

Model training

[Access trained AI model](#)

Access trained AI model

- Once training is complete, you can find your AI models within the `models/` directory (choose one for the following AI model deployment step)



-
- 1. AI model training.
 - 2. AI model deployment
 - 3. AI model analysis
 - 4. AI model integration

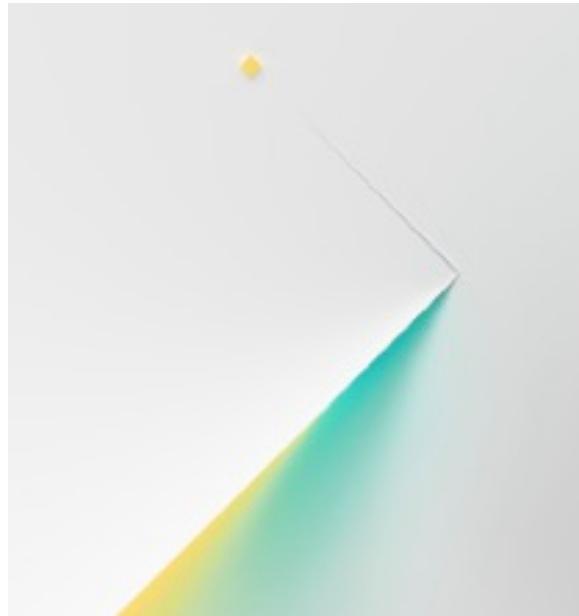
AI model training complete

1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration



Environments used

Windows

Linux on Z

Step 2

AI model deployment

We will deploy our fraud detection AI model using MLz. We can utilize the model import functionality on the MLz UI. This deployed AI model can then be integrated into applications within the IBM Z environment.

1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

[Access sample code](#)

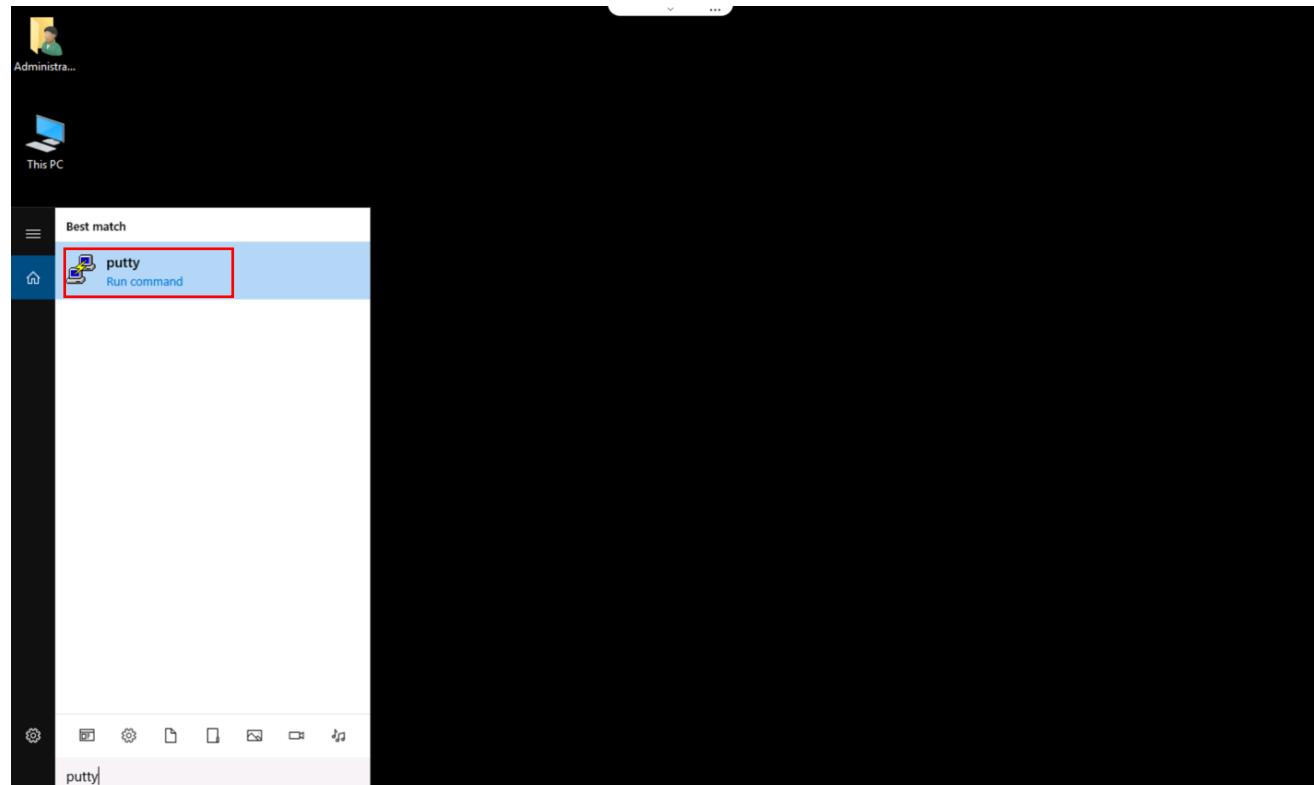
[Integrate AI model into
Triton Inference Server](#)

[Deploy Triton Inference
Server](#)

[Run sample test](#)

Access sample code

1. Open PuTTY application



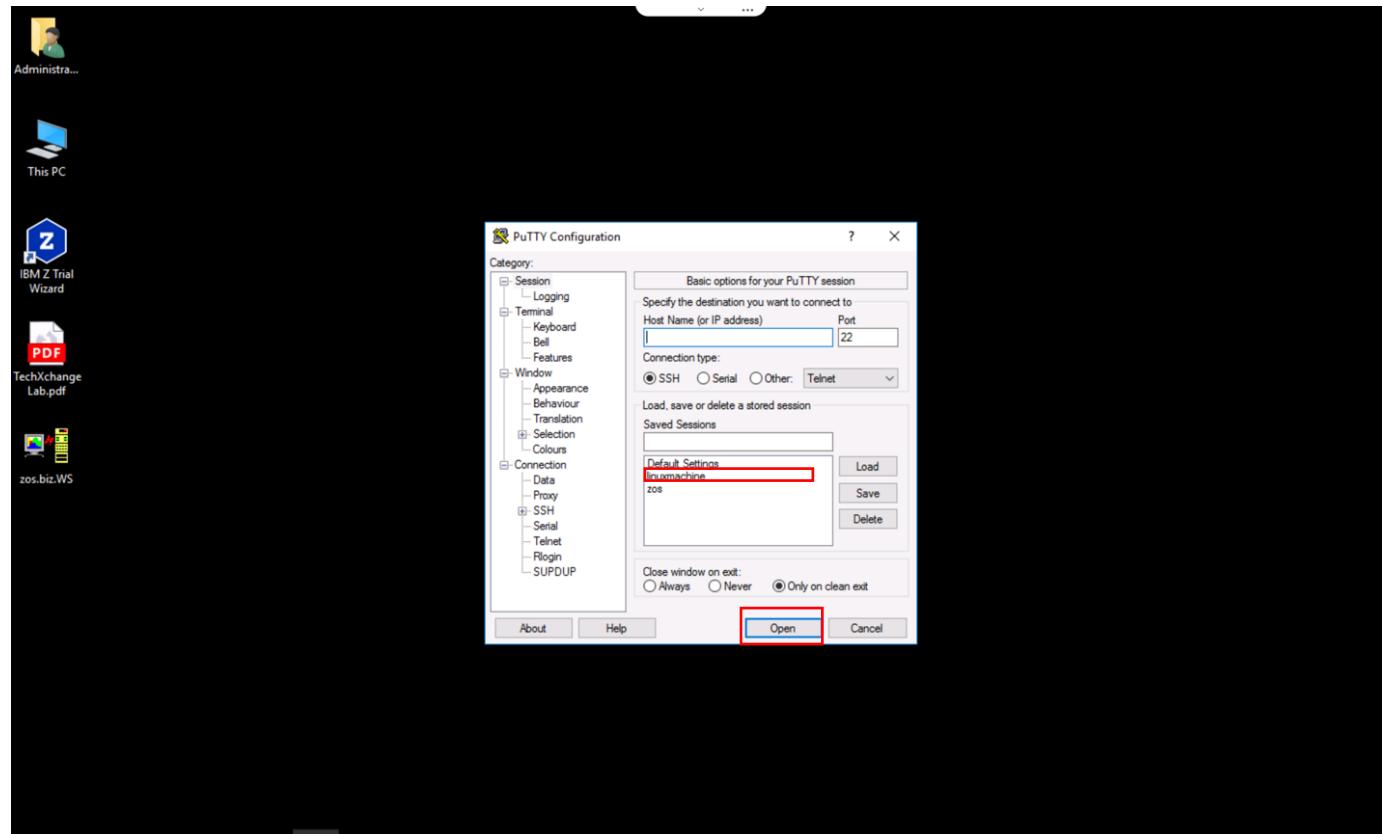
1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

2. Double click on linuxmachine



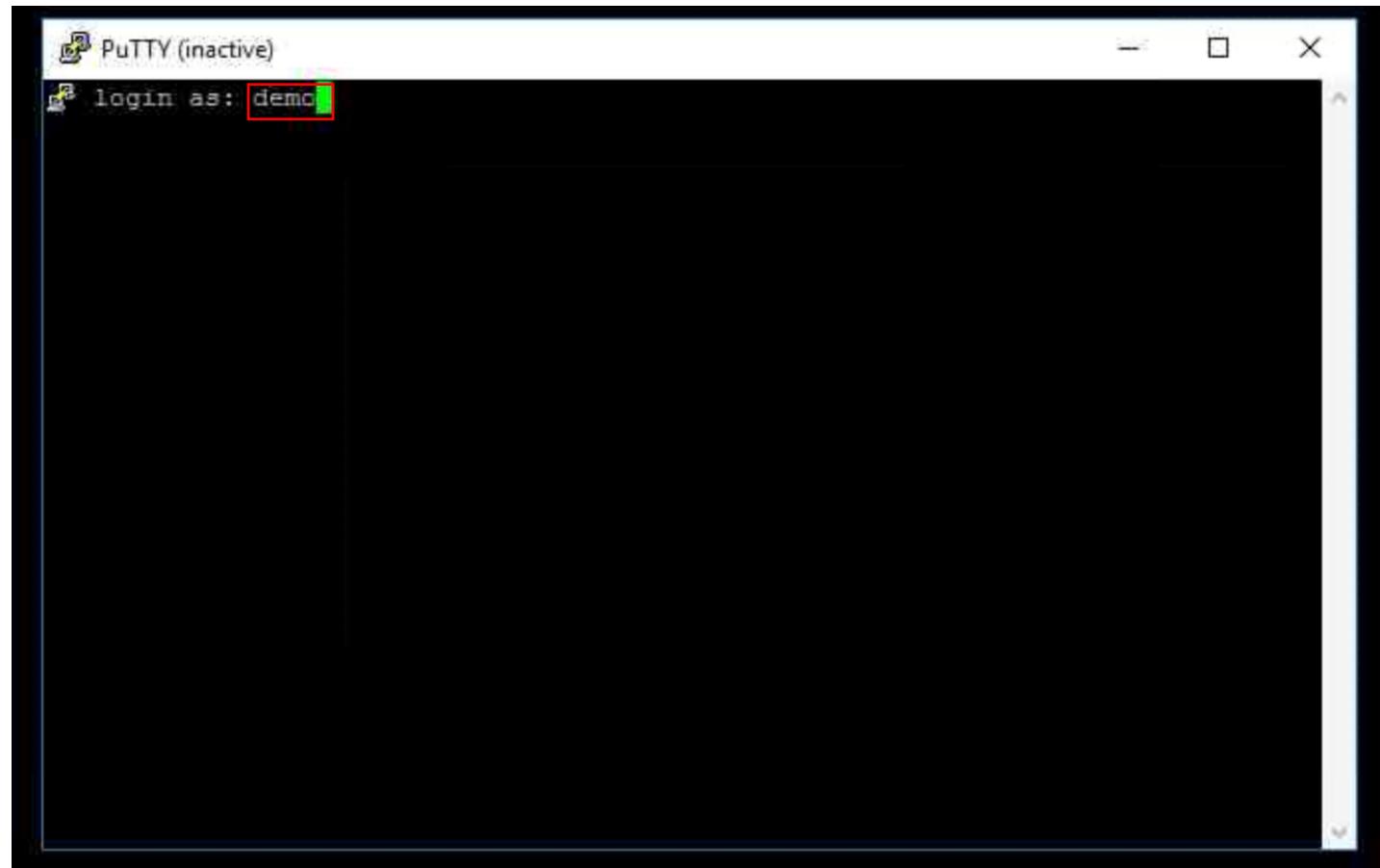
1. AI model training.

2. AI model deployment

3. AI model analysis

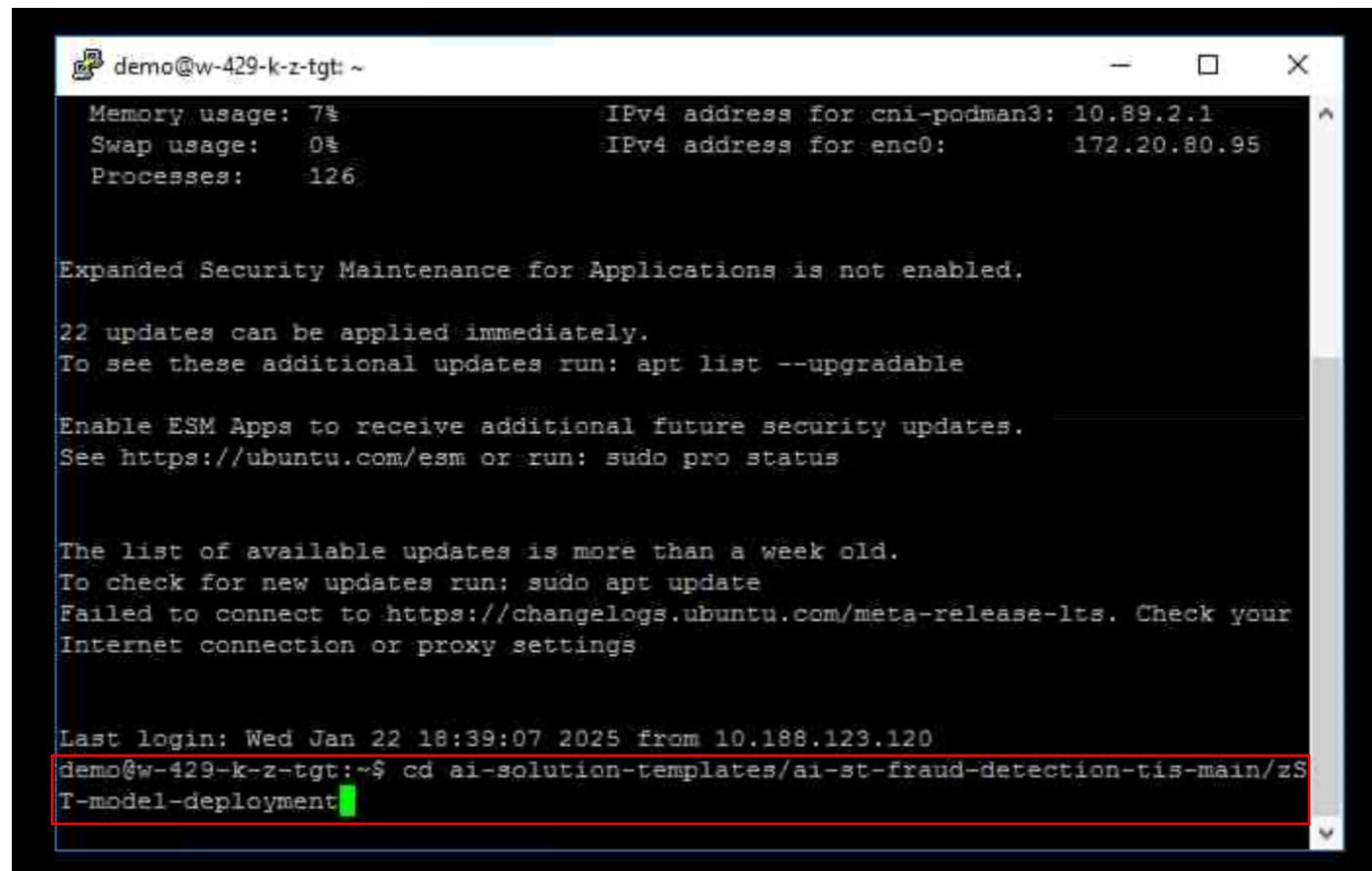
4. AI model integration

3. Enter demo username



4. Move to sample code directory

```
cd ai-solution-templates/ai-st-fraud-detection-tis-
main/zST-model-deployment
```



A screenshot of a terminal window titled "demo@w-429-k-z-tgt: ~". The window displays various system statistics and a command prompt. The command "cd ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model-deployment" is highlighted with a red rectangle at the bottom of the terminal window.

```
demo@w-429-k-z-tgt: ~
Memory usage: 7%
Swap usage: 0%
Processes: 126
IPv4 address for cni-podman3: 10.89.2.1
IPv4 address for ens0: 172.20.80.95

Expanded Security Maintenance for Applications is not enabled.
22 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Wed Jan 22 18:39:07 2025 from 10.188.123.120
demo@w-429-k-z-tgt:~$ cd ai-solution-templates/ai-st-fraud-detection-tis-main/zS
T-model-deployment
```

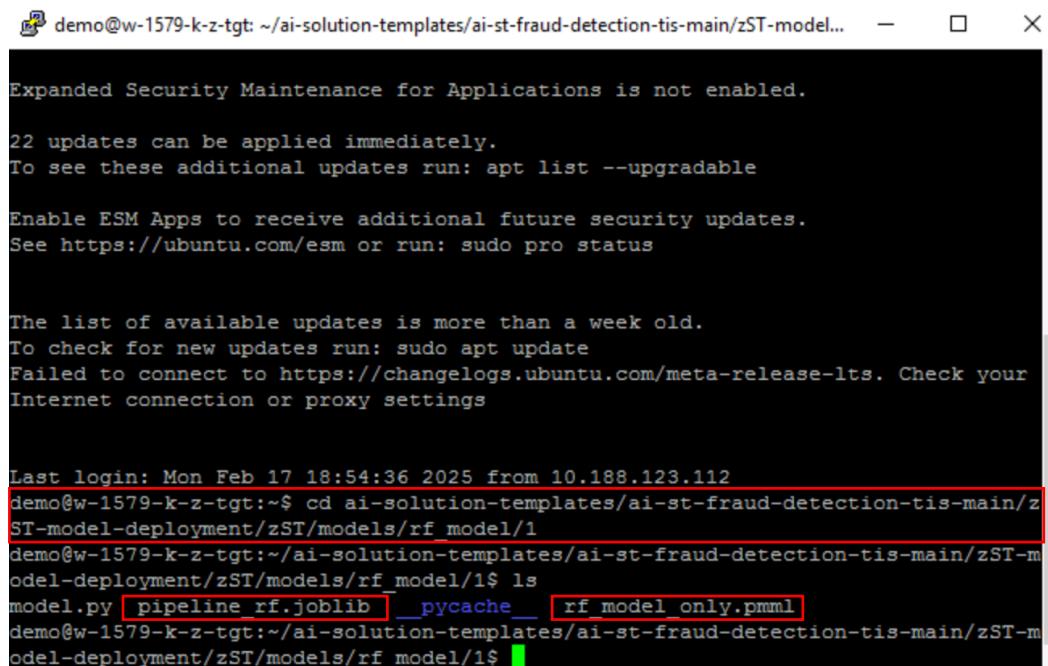
Integrate AI model into Triton Inference Server

1. Add your model (.pmml file) to

```
~/ai-solution-templates/ai-st-fraud-detection-tis-
main/zST-model-
deployment/zST/models/rf_model/1 directory
```

2. Add your preprocessing .joblib file to

```
~/ai-solution-templates/ai-st-fraud-detection-tis-
main/zST-model-
deployment/zST/models/rf_model/1 directory
```



A terminal window showing a Linux command-line interface. The window title is "demo@w-1579-k-z-tgt: ~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model...". The terminal displays several messages:

- System security update information: "Expanded Security Maintenance for Applications is not enabled.", "22 updates can be applied immediately.", "To see these additional updates run: apt list --upgradable", "Enable ESM Apps to receive additional future security updates.", "See https://ubuntu.com/esm or run: sudo pro status".
- Information about available updates: "The list of available updates is more than a week old.", "To check for new updates run: sudo apt update", "Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings".
- User navigation and file listing:

```
demo@w-1579-k-z-tgt:~$ cd ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model-deployment/zST/models/rf_model/1
demo@w-1579-k-z-tgt:~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model-deployment/zST/models/rf_model/1$ ls
model.py pipeline rf.joblib pycache rf_model_only.pmml
```

Access sample code

Integrate AI model into
Triton Inference Server

[Deploy Triton Inference
Server](#)

Run sample test

Deploy Triton Inference Server

1. Run podman container

```
podman run --net=my-data-network --shm-size 1G -u root --rm -p8000:8000 --name=zst-tis-app -v//$PWD/zST/models:/models zst-tis tritonserver --model-repository=/models
```

The screenshot shows a terminal window with a black background and white text. The command entered is:

```
demo@w-429-k-z-tgt:~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model-deployment$ podman run --net=my-data-network --shm-size 1G -u root --rm -p8000:8000 --name=zst-tis-app -v//$PWD/zST/models:/models zst-tis tritonserver --model-repository=/models
```

Below the command, the terminal displays several log messages from the Triton server:

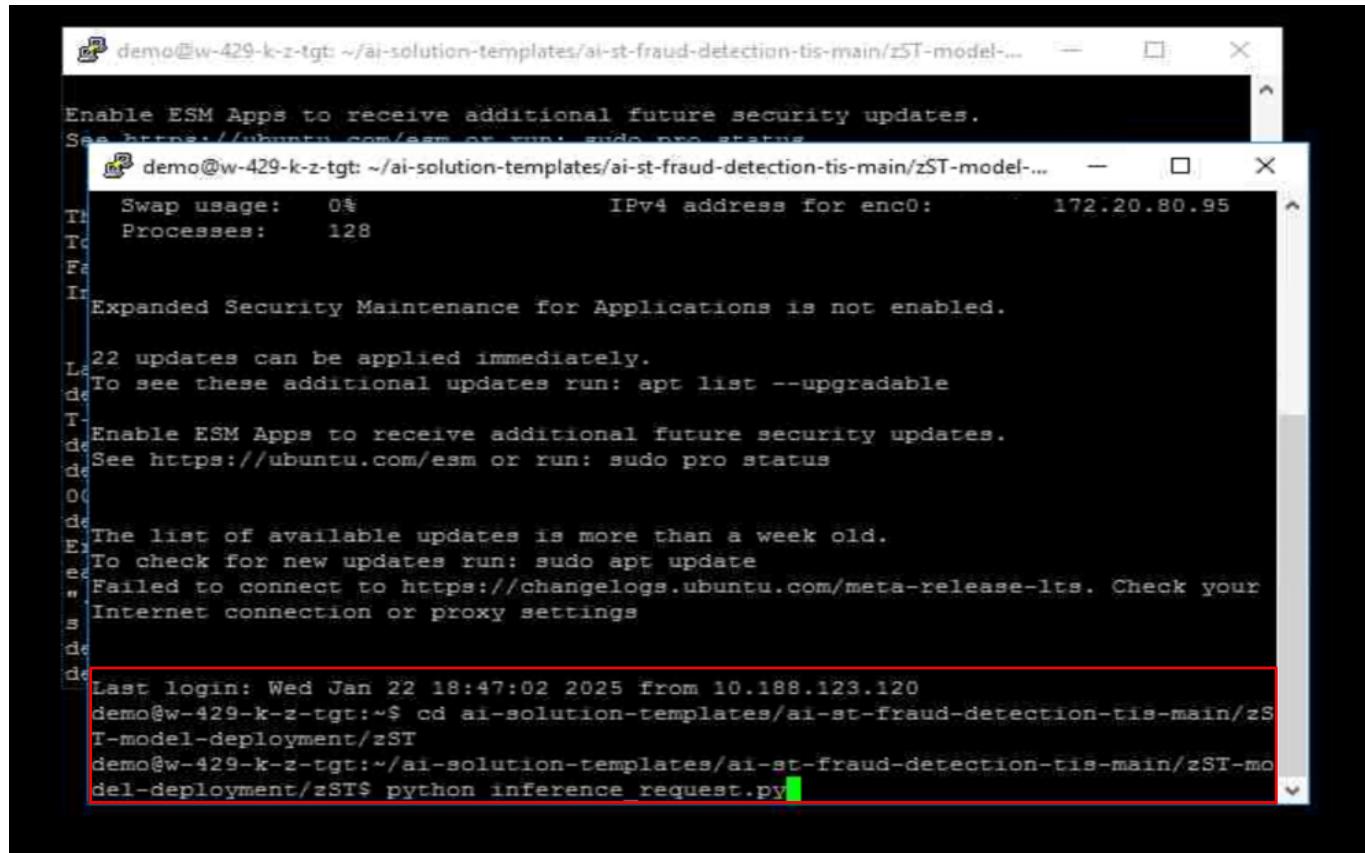
```
I0122 18:34:04.389045 1 model_lifecycle.cc:459] loading: rf_model:1
I0122 18:34:10.304987 1 python_be.cc:1977] TRITONBACKEND_ModelInstanceInitialize : rf_model_0_0 (CPU device 0)
I0122 18:34:11.379244 1 python_be.cc:1977] TRITONBACKEND_ModelInstanceInitialize : rf_model_0_1 (CPU device 0)
I0122 18:34:12.424050 1 model_lifecycle.cc:694] successfully loaded 'rf_model' version 1
I0122 18:34:12.424179 1 server.cc:583]
+---+---+
| Repository Agent | Path |
+---+---+
+---+---+
I0122 18:34:12.424235 1 server.cc:610]
+---+---+
+---+---+
| Backend | Path | Config
+---+---+
```

Run sample test

1. Run python script from terminal with ip/port of triton inference server (in new terminal)

```
cd ~/ai-solution-templates/ai-st-fraud-detection-tis-
main/zST-model-deployment/zST
```

```
python inference_request.py
```



```
demo@w-429-k-z-tgt:~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model-...
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
Swap usage:  0%          IPv4 address for eno:      172.20.80.95
Processes:   128
In
Expanded Security Maintenance for Applications is not enabled.

22 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings
Last login: Wed Jan 22 18:47:02 2025 from 10.188.123.120
demo@w-429-k-z-tgt:~$ cd ai-solution-templates/ai-st-fraud-detection-tis-main/zS
T-model-deployment/zST
demo@w-429-k-z-tgt:~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-mo
del-deployment/zSTS$ python inference request.py
```

1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

AI model deployment complete



Environments used

Linux on Z
Windows

Step 3

AI model analysis

We will analyze our fraud detection AI model using a sample AI on Linux on Z Fraud Detection Dashboard. We can invoke the API of this sample dashboard from another sample application to visualize the AI model inferencing.

All sample code for this section is within

```
ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model-analysis
```

1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

[Access sample code](#)

[Configure sample application](#)

[Build sample application](#)

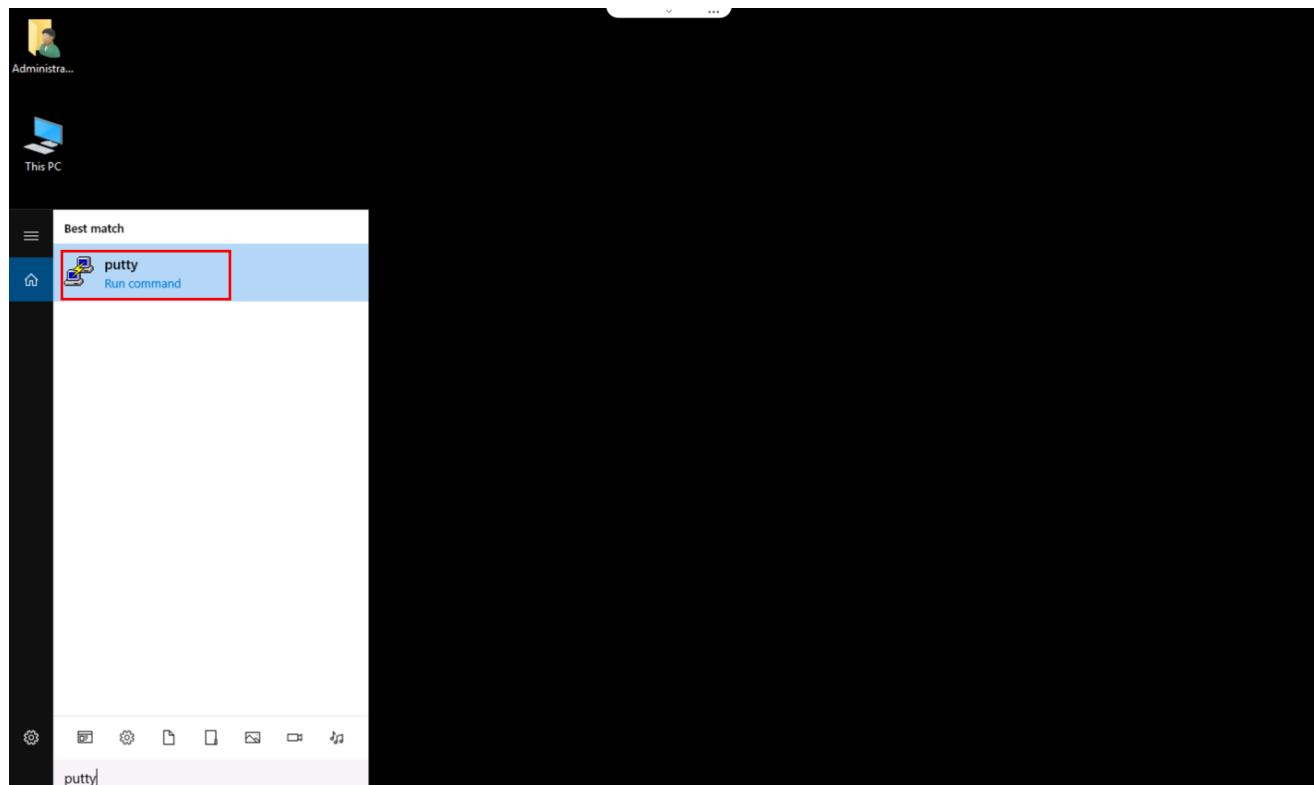
[Access sample application](#)

[Analyze credit card events](#)

[Make predication](#)

Access sample code

1. Open PuTTY application



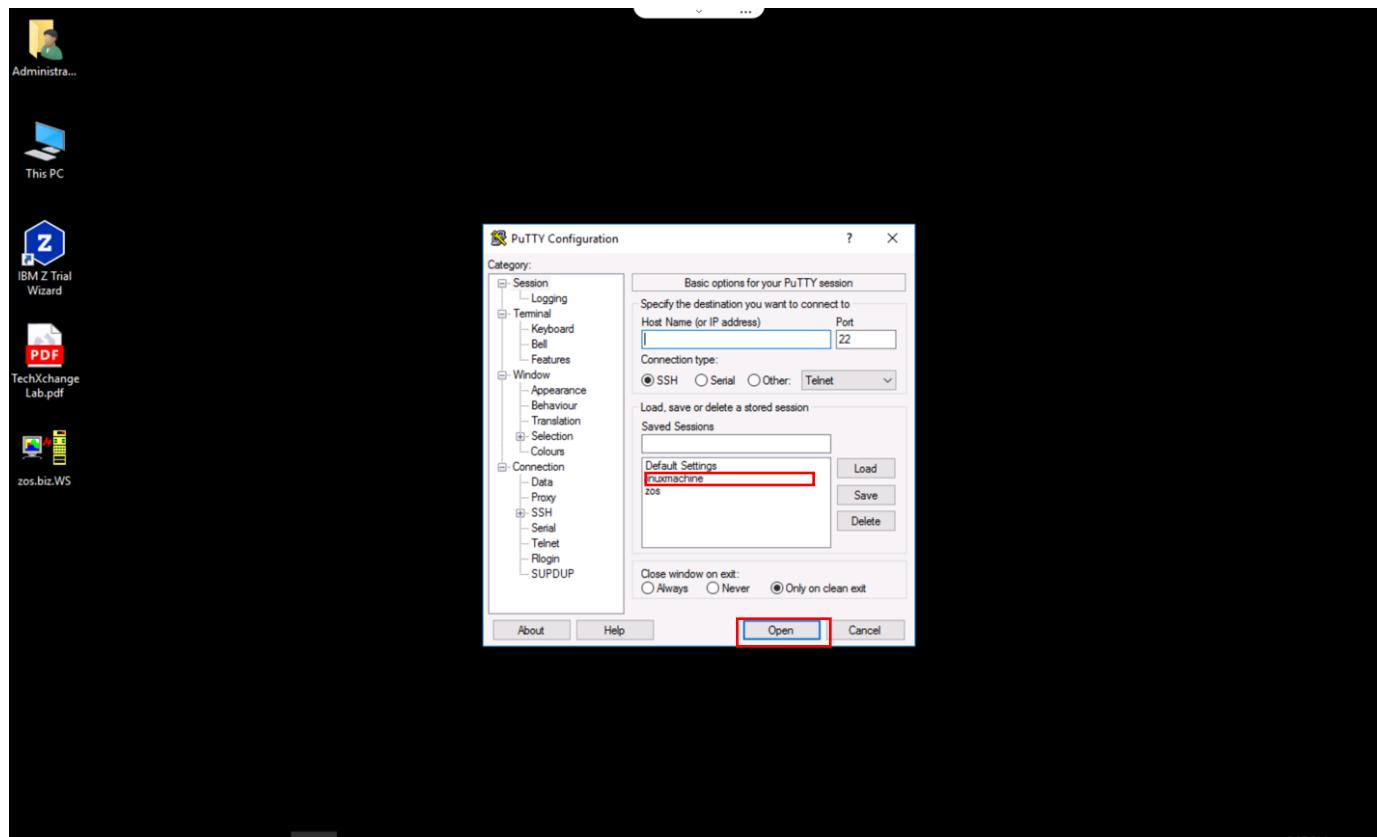
1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

2. Double click on linuxmachine



1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

3. Enter demo username



1. AI model training.

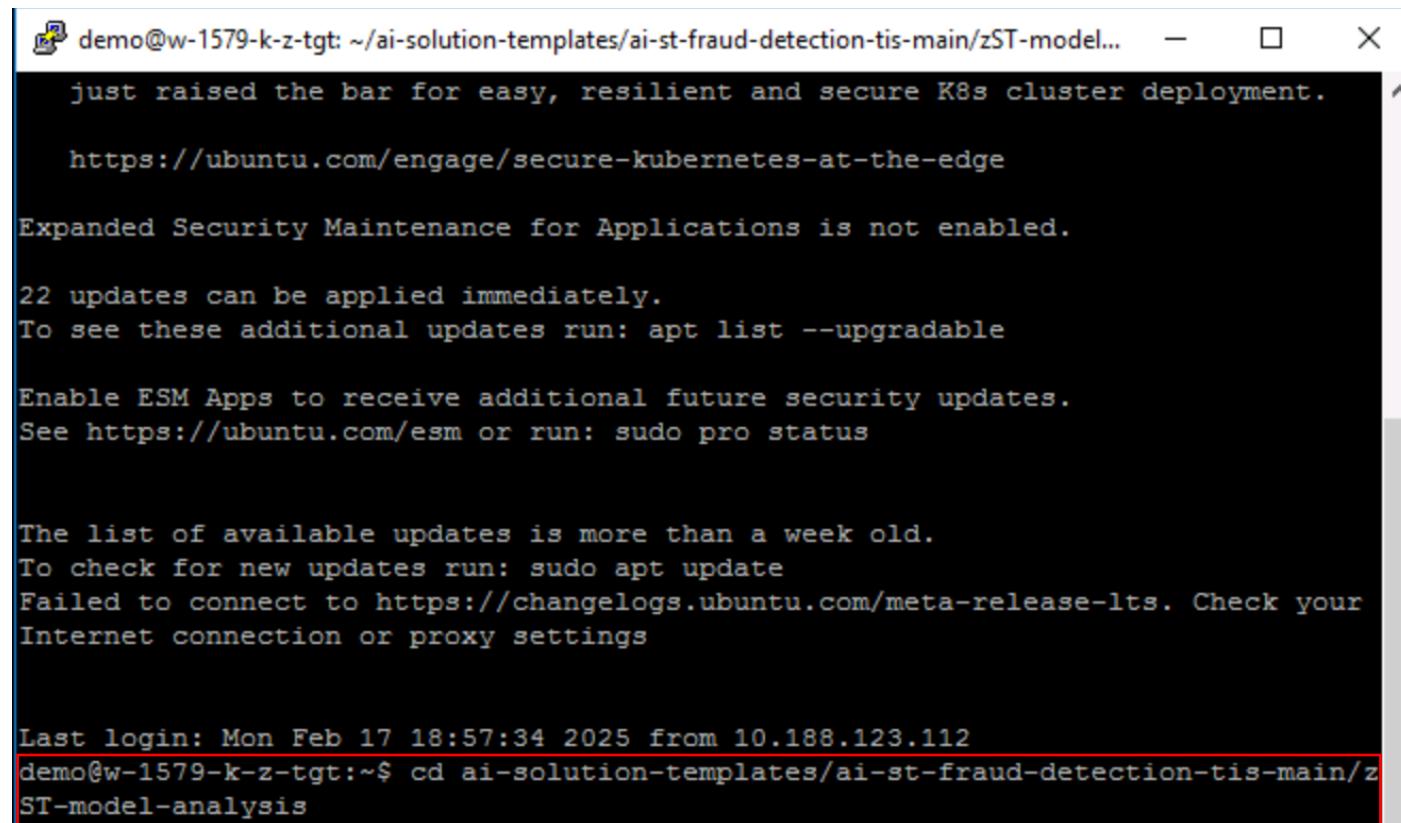
2. AI model deployment

3. AI model analysis

4. AI model integration

4. Move to sample code directory

```
cd ai-solution-templates/ai-st-fraud-detection-tis-
main/zST-model-analysis
```



The screenshot shows a terminal window titled "demo@w-1579-k-z-tgt: ~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model-analysis". The window displays several lines of text related to system updates and maintenance. At the bottom of the window, the command "cd ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model-analysis" is visible, highlighted with a red rectangle.

```
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

22 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

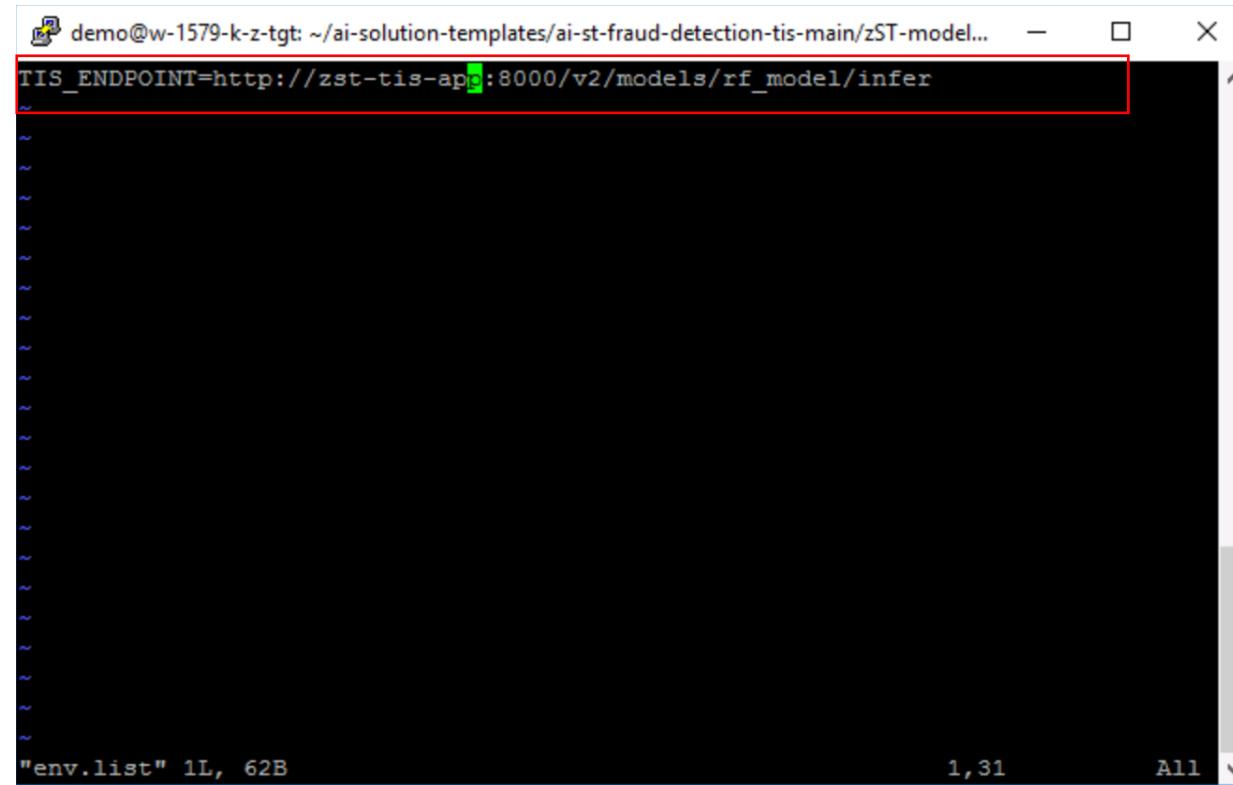
Last login: Mon Feb 17 18:57:34 2025 from 10.188.123.112
demo@w-1579-k-z-tgt:~$ cd ai-solution-templates/ai-st-fraud-detection-tis-main/z
ST-model-analysis
```

Configure sample application

1. Set the environment variables within

```
~/ai-solution-templates/ai-st-fraud-detection-tis-  
main/zST-model-analysis/env.list
```

TIS_ENDPOINT (scoring endpoint for deployed AI
model)



A screenshot of a terminal window titled "demo@w-1579-k-z-tgt: ~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model...". The window shows a single command being entered:

```
TIS_ENDPOINT=http://zst-tis-app:8000/v2/models/rf_model/infer
```

The terminal interface includes standard navigation keys (~, ^D, ^S, ^W, ^E) and status indicators at the bottom: "env.list" 1L, 62B, 1, 31, All.

Get model details for inferencing

Access sample code

Configure sample application

[Deploy sample application](#)

[Access sample application](#)

Analyze credit card events

Make prediction

Deploy sample application

1. Run command in terminal (e.g. port 5002)

```
podman run --rm -p 5002:5002 --net=my-data-network --env-file env.list --name model-analysis-tis-app model-analysis-tis
```

The screenshot shows a terminal window with two tabs. The left tab displays configuration options for a Triton server, including server_id (triton), server_version, and various extensions like nvidia and shared_memory. The right tab shows the command being run:

```
demo@t-19807-k-z-tgt:~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-model-deployment$ podman run --rm -p 5002:5002 --net=my-data-network --env-file env.list --name model-analysis-tis-app model-analysis-tis
```

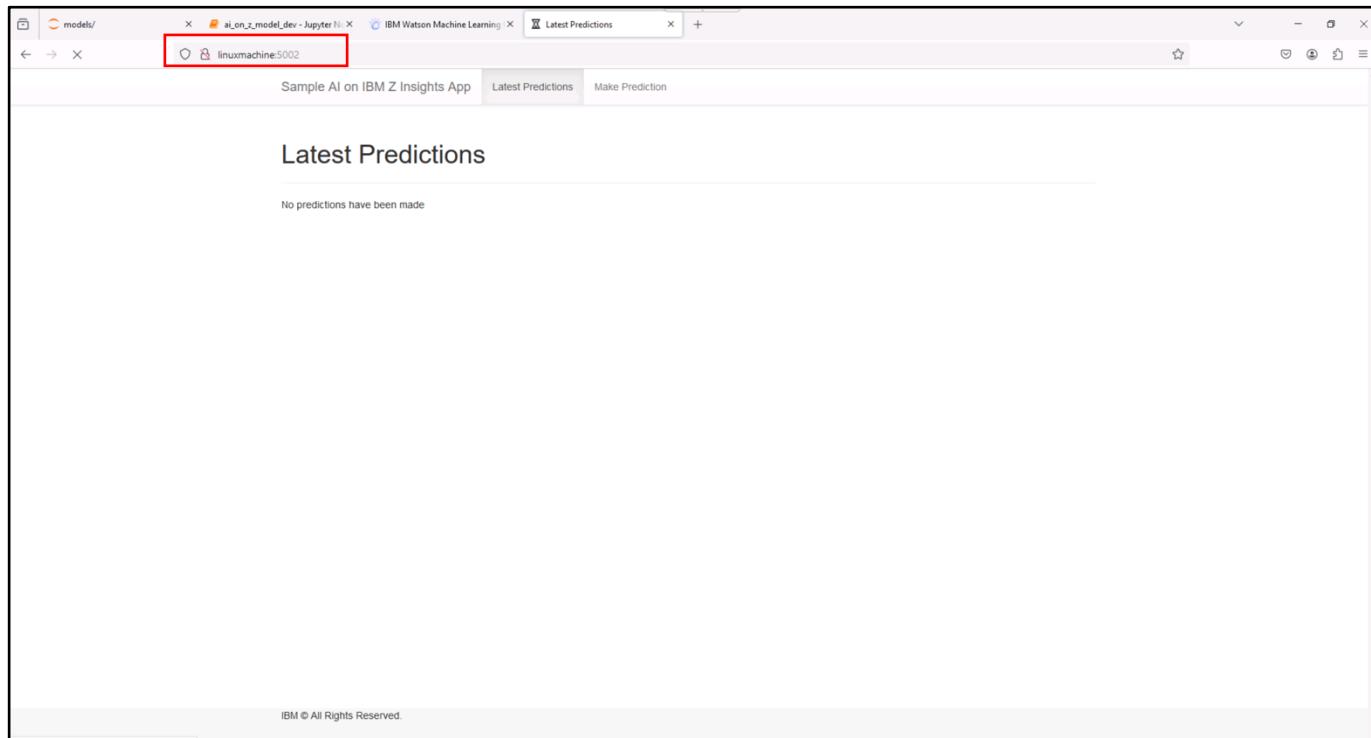
The output of the command shows the server starting up on port 5002, serving a Flask app, and running in debug mode. It also warns against using it in production and lists the addresses it's listening on: 0.0.0.0, 127.0.0.1, and 10.89.1.3. A red box highlights the command line in the terminal.

Access sample application

1. View the following URL in a web browser

<http://linuxMachine:{port}>

- a. Port: port you used with podman run



1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

Get model details for
inferencing

Access sample code

Configure sample
application

Deploy sample application

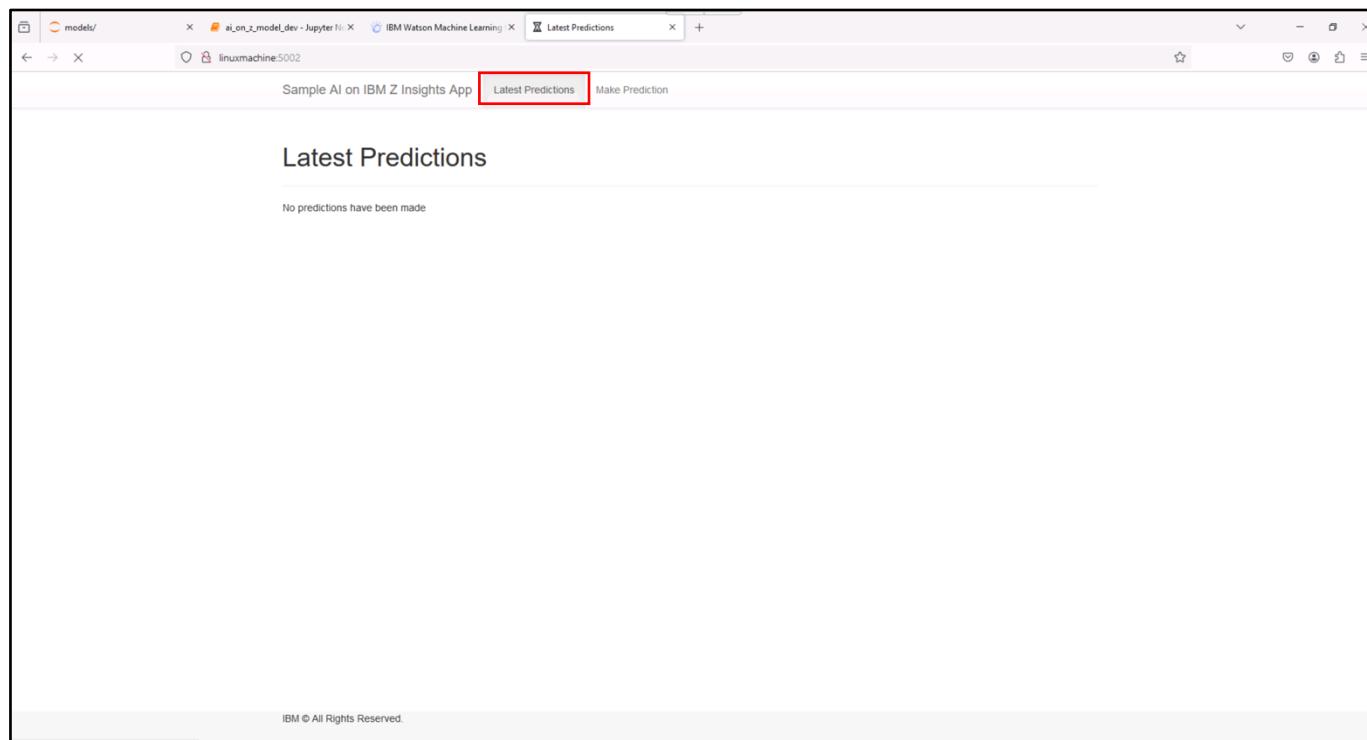
Access sample application

Analyze credit card events

Make prediction

Analyze credit card events

1. Go to latest predictions tab



1. AI model training.

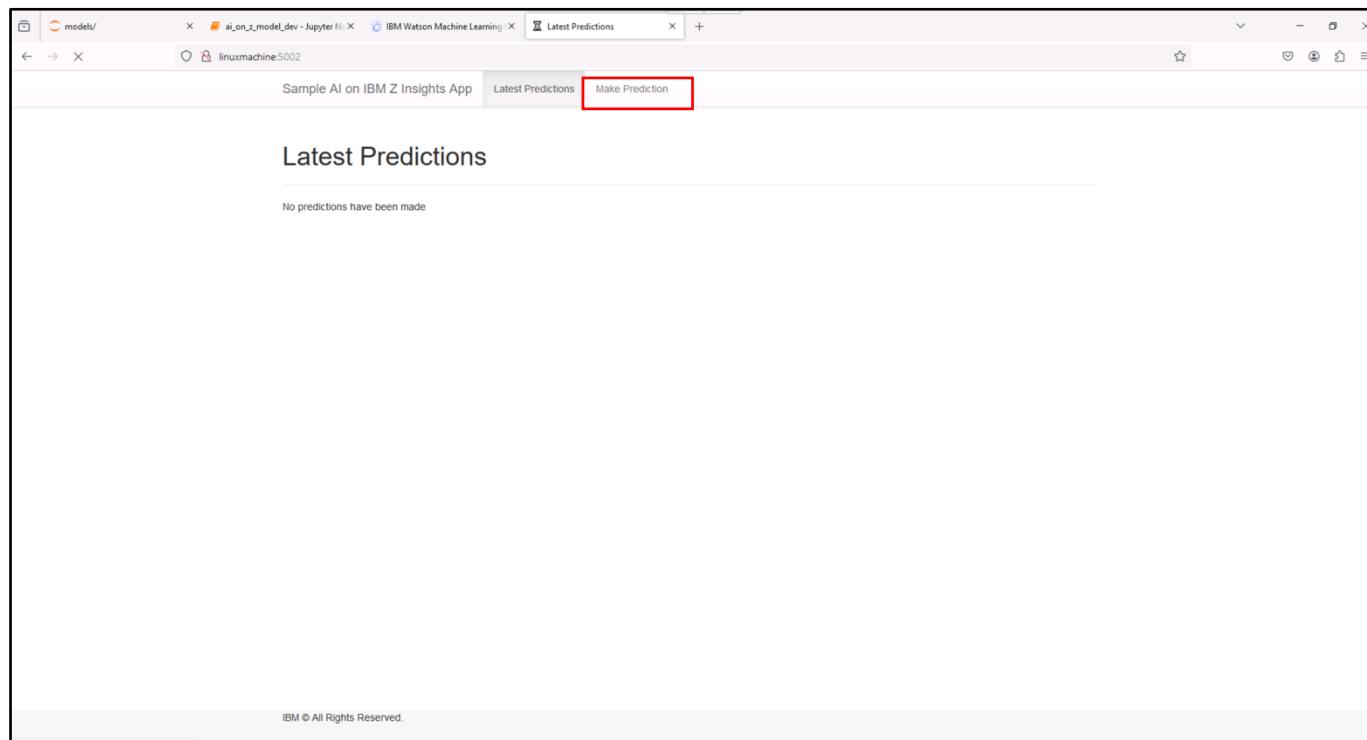
2. AI model deployment

3. AI model analysis

4. AI model integration

Make predication

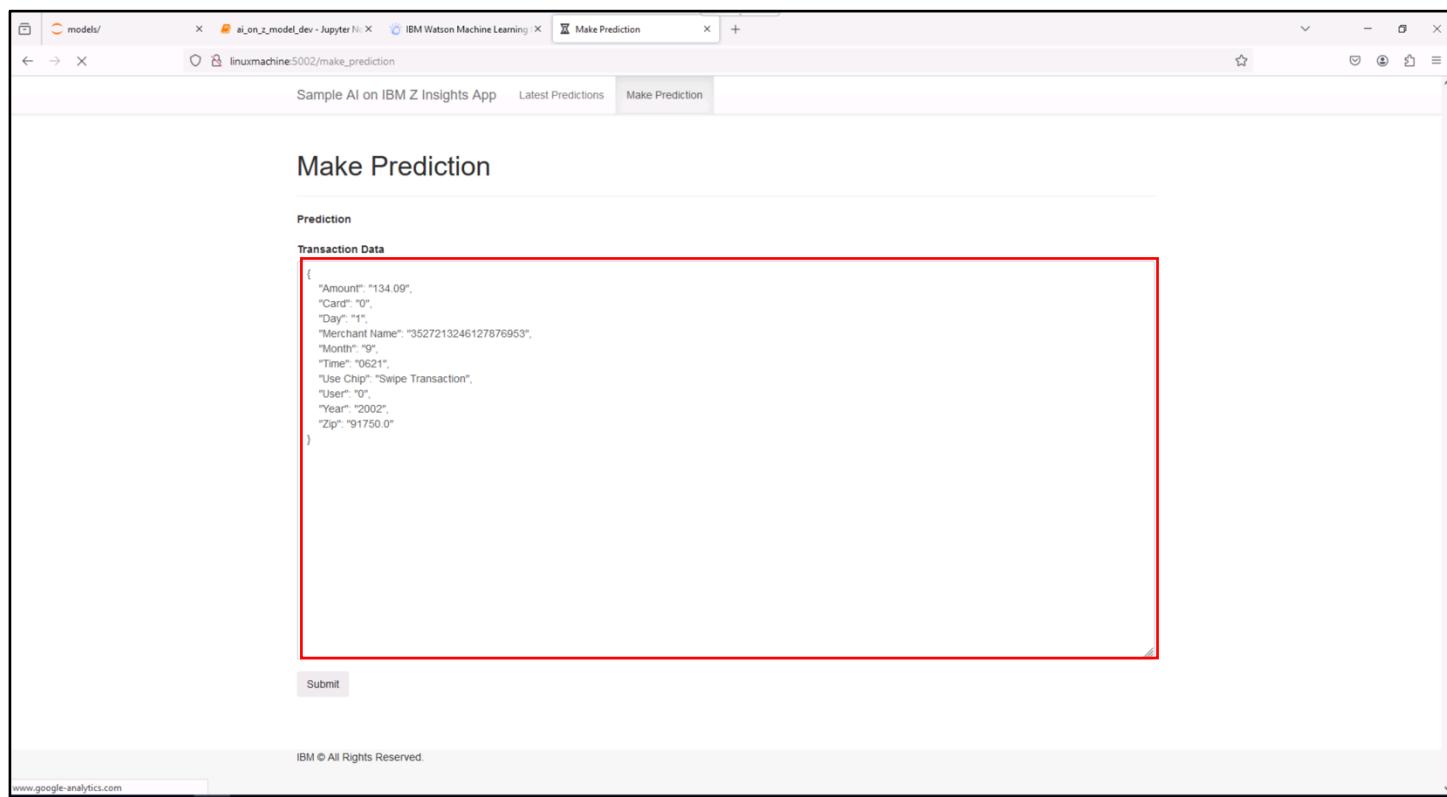
1. Go to latest make predication tab



2. Input json data in following format:

a.

```
{"Amount": "134.09", "Card": "0", "Day": "1",
"Merchant Name":
"3527213246127876953", "Month": "9",
"Time": "0621", "Use Chip": "Swipe
Transaction", "User": "0", "Year": "2002",
"Zip": "91750.0"}
```



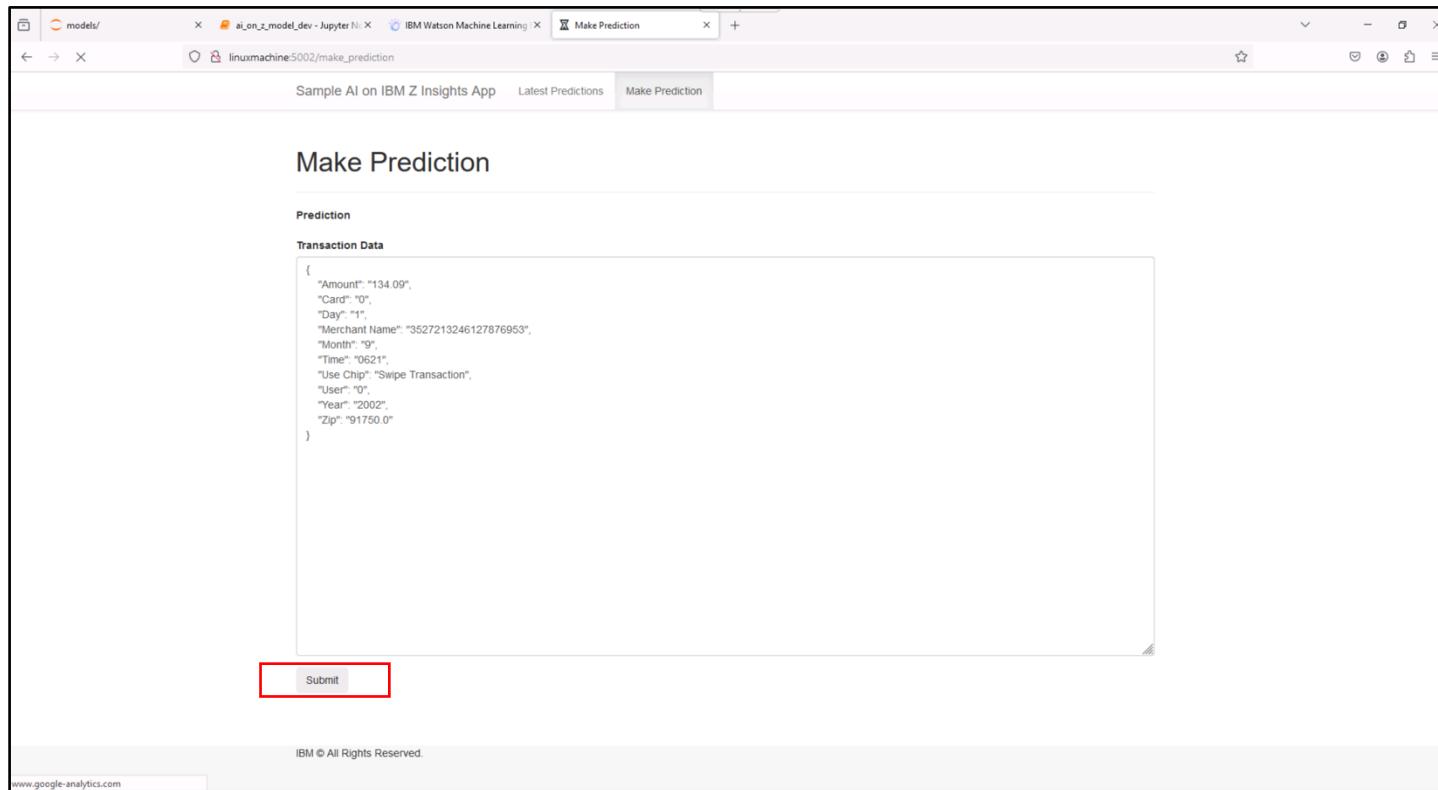
1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

3. Click submit



1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

AI model analysis complete



Environments used

Linux on Z

Windows

Prerequisites

- Must have AI on IBM Z Sample Fraud Detection Dashboard deployed for inferencing and analysis
- Must have Docker or Podman installed

Step 4

AI model integration

[Temporarily Removed]

We can use our deployed TIS fraud detection AI model and integrate it into different types of applications. The AI model can be analyzed and/or provide inferencing APIs using the sample AI on Linux on Z Fraud Detection Dashboard.

All sample code for this section is within

`ai-solution-templates/ai-st-fraud-detection-tis-main/zST-storefront-evershop`

1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

[Access sample code](#)

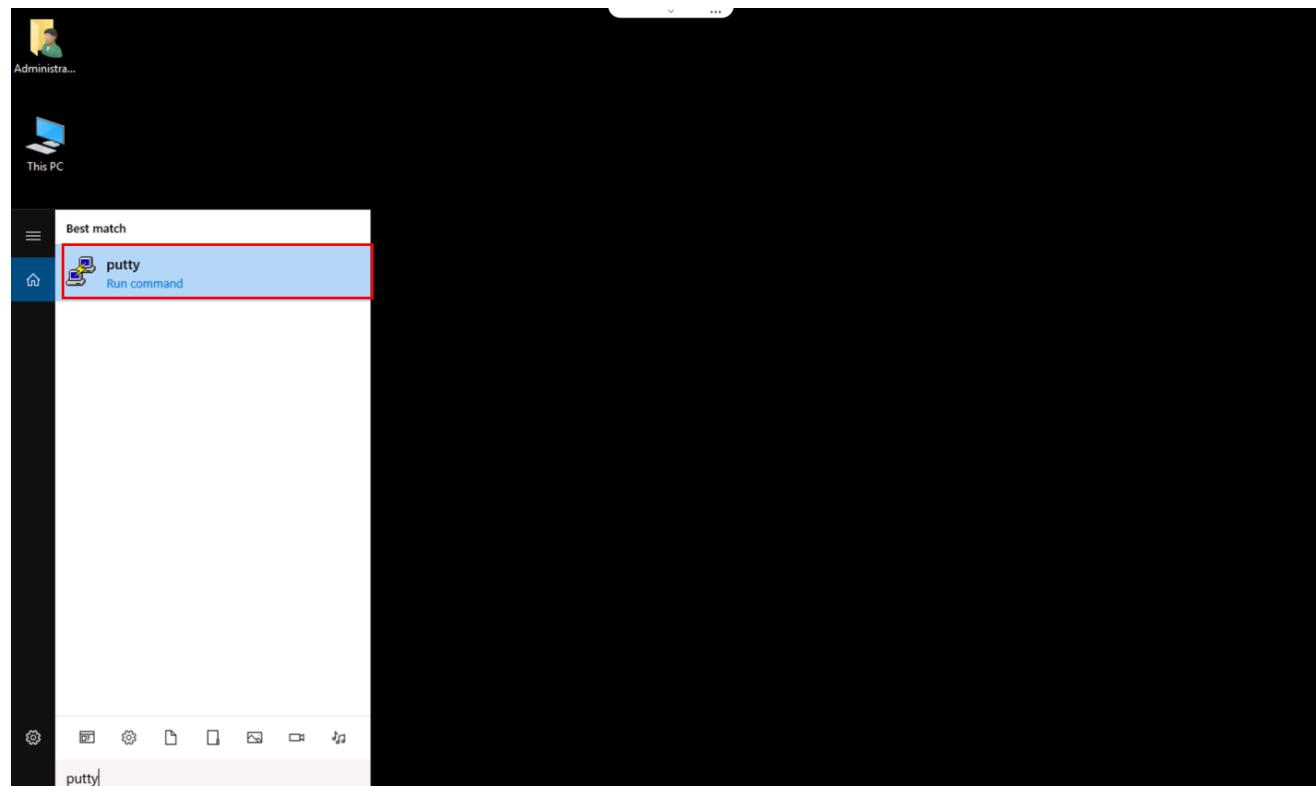
[Deploy sample ecommerce application](#)

[Access sample ecommerce application](#)

[Use fraud detection AI model with EverShop Storefront](#)

Access sample code

1. Open PuTTY application



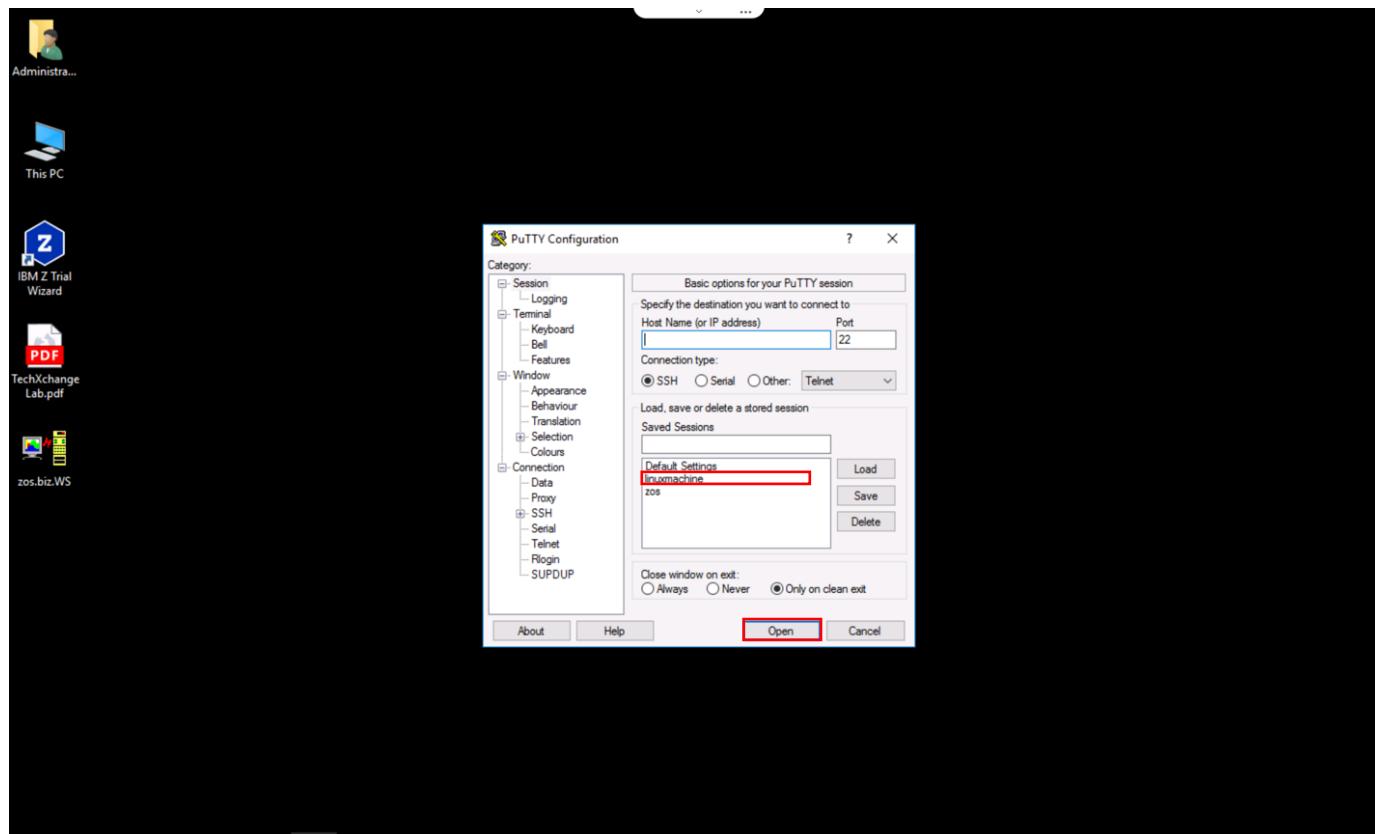
1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

2. Double click on **linuxmachine**



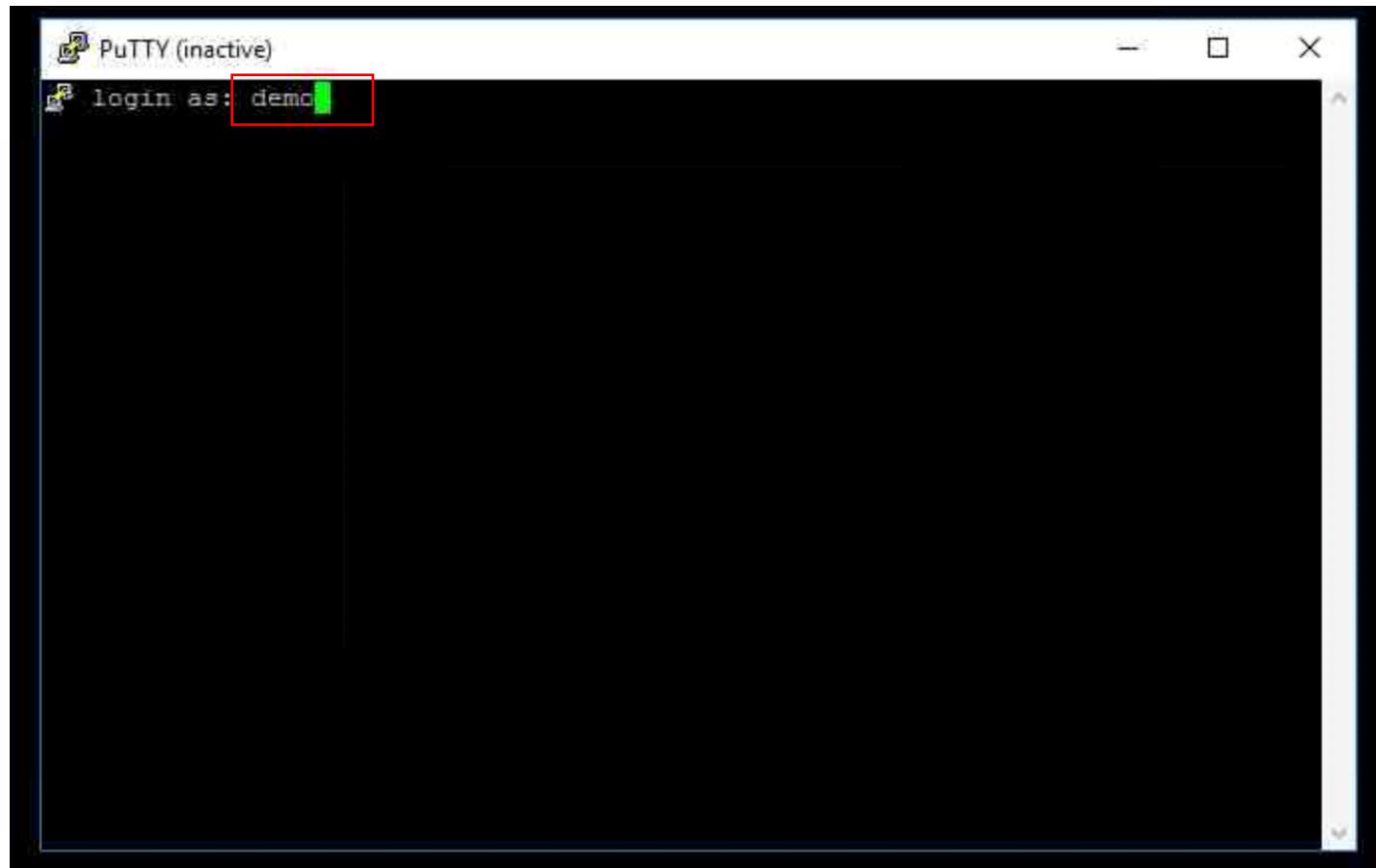
1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

3. Enter demo username



1. AI model training.

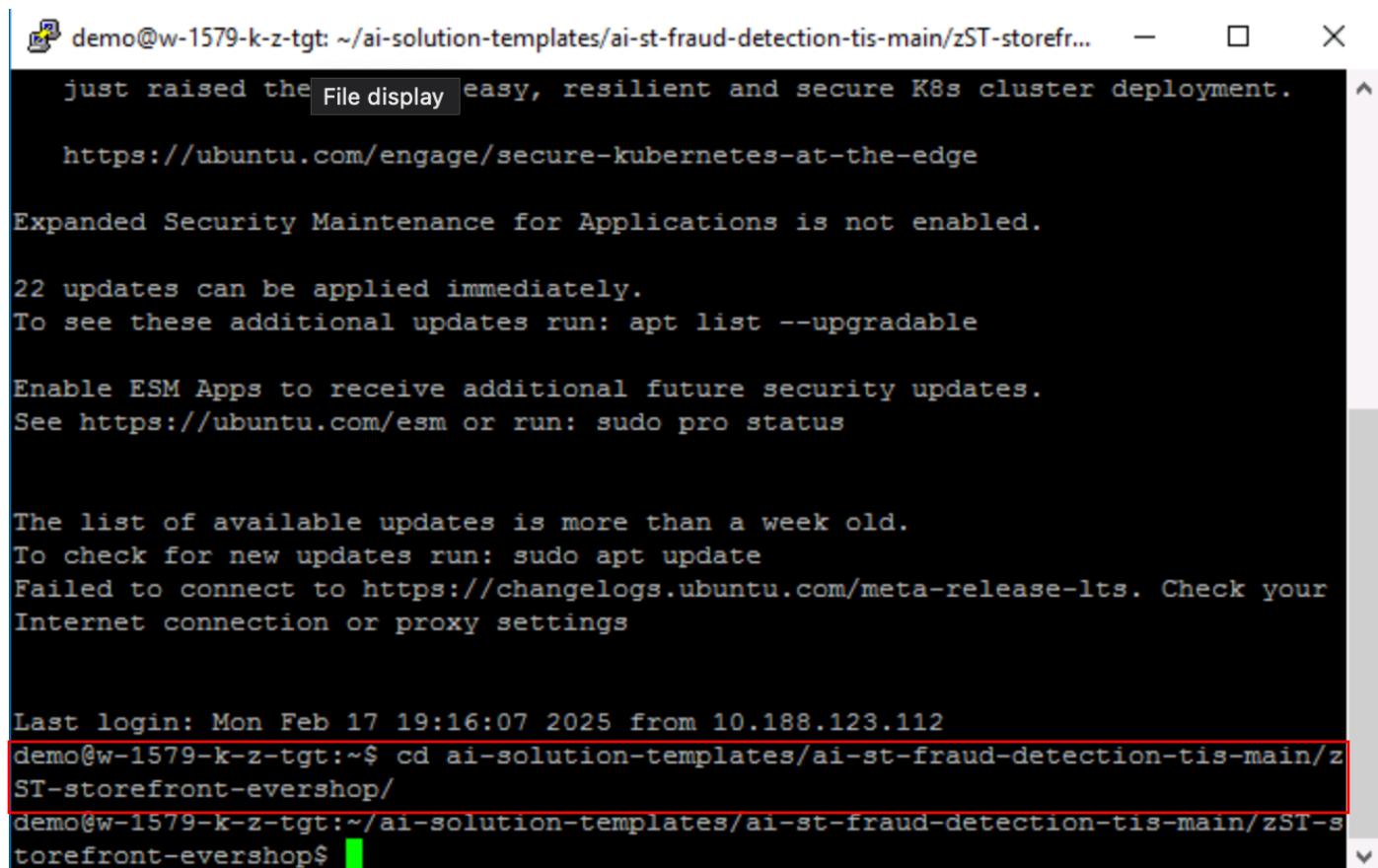
2. AI model deployment

3. AI model analysis

4. AI model integration

4. Move to sample code directory

```
cd ai-solution-templates/ai-st-fraud-detection-tis-
main/zST-storefront-evershop
```



A terminal window titled 'demo@w-1579-k-z-tgt: ~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-storefr...' displays the output of a command. The output includes a message about easy, resilient, and secure K8s cluster deployment, a link to secure Kubernetes at the edge, information about ESM Apps, and a warning about old available updates. The last part of the output shows the user navigating to the directory 'ai-solution-templates/ai-st-fraud-detection-tis-main/zST-storefront-evershop/'.

```
just raised the File display easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

22 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

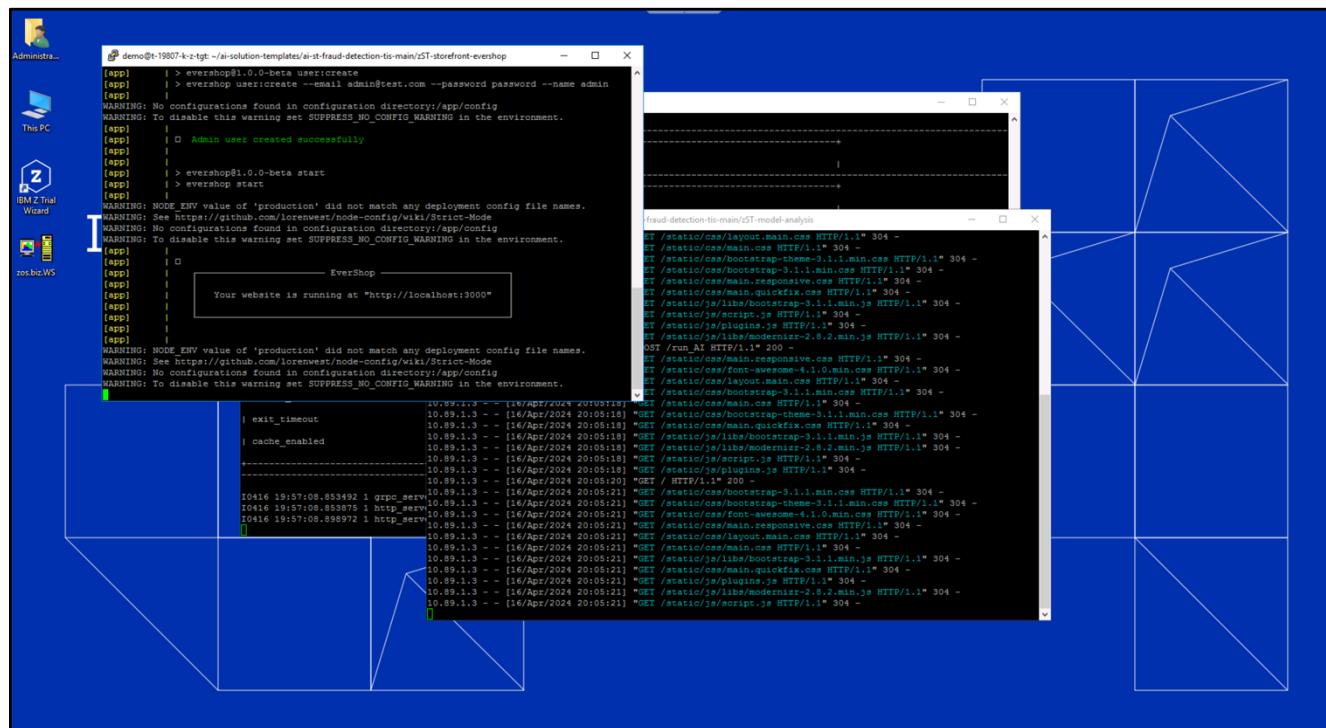
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Mon Feb 17 19:16:07 2025 from 10.188.123.112
demo@w-1579-k-z-tgt:~$ cd ai-solution-templates/ai-st-fraud-detection-tis-main/z
ST-storefront-evershop/
demo@w-1579-k-z-tgt:~/ai-solution-templates/ai-st-fraud-detection-tis-main/zST-s
torefront-evershop$
```

Deploy sample ecommerce application

- #### 1. Run command in terminal (password: demo)

```
sudo podman-compose up --no-build
```



1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

Access sample code

Deploy sample
ecommerce application

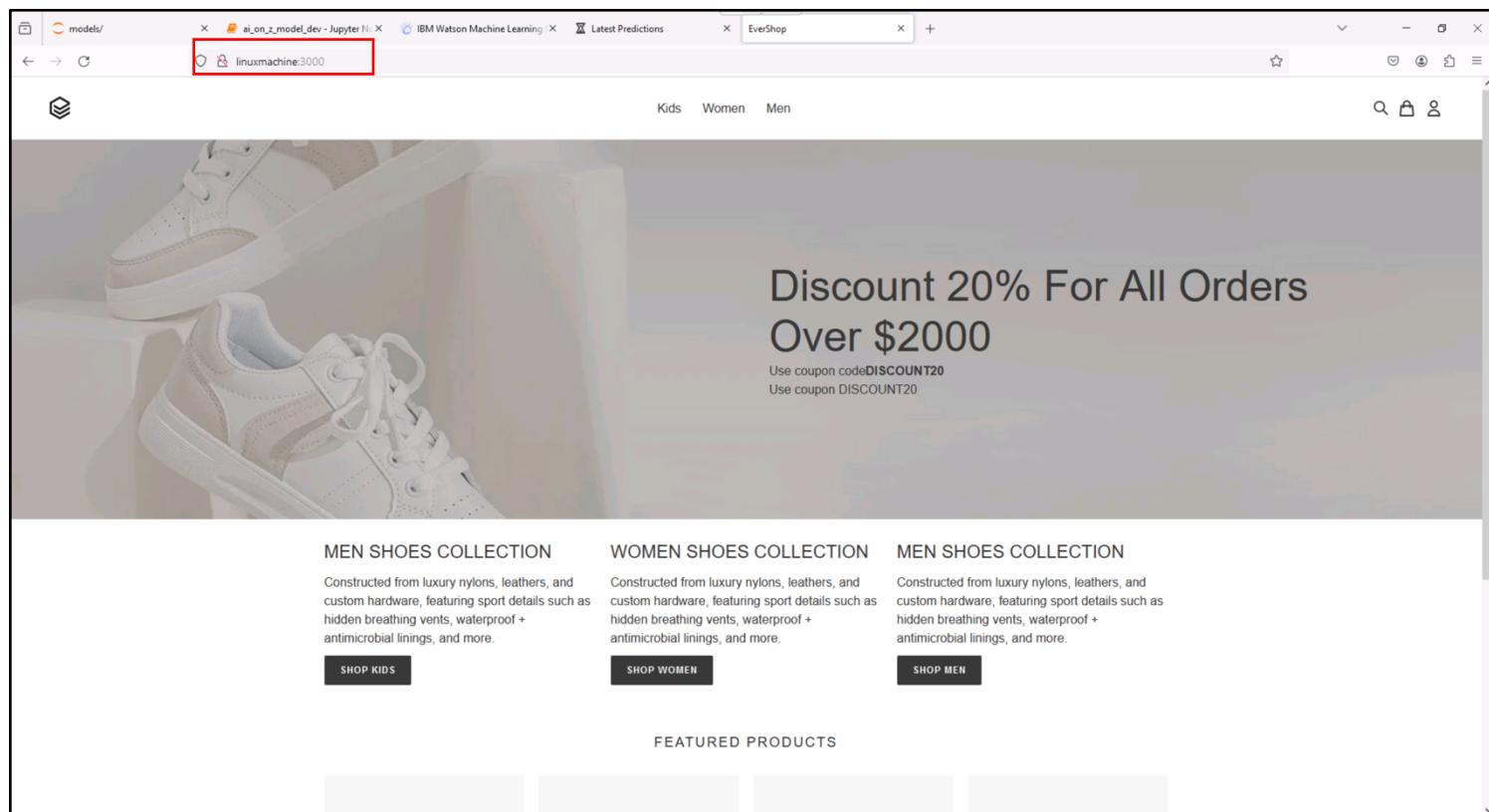
Access sample
ecommerce application

Use fraud detection AI
model with EverShop
Storefront

Access sample ecommerce application

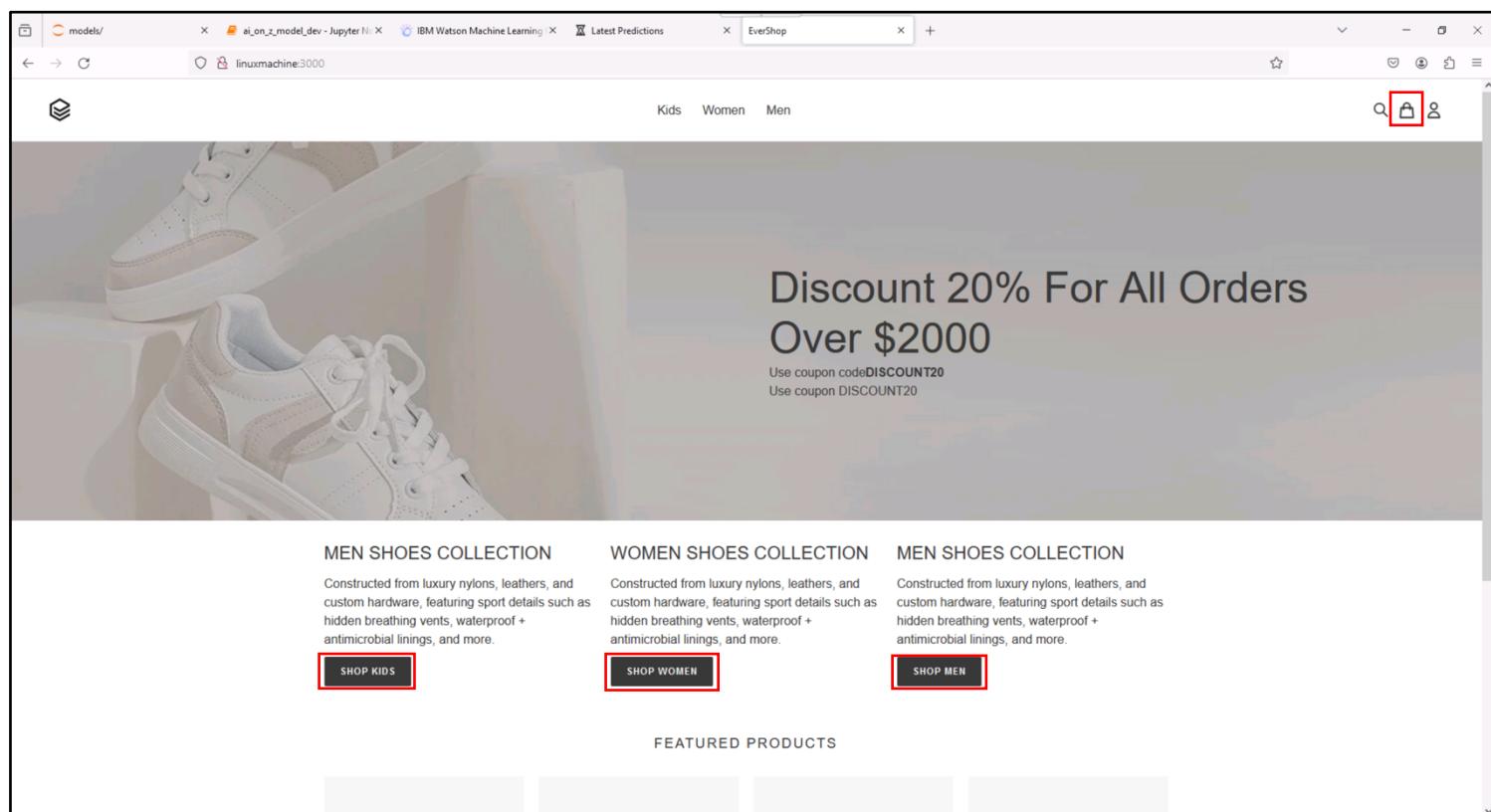
1. Enter URL in web browser using app URL

<http://linuxMachine:3000>



Use fraud detection AI model with EverShop Storefront

1. Make sure you have AI on IBM Z Sample Fraud Detection Dashboard deployed for inferencing and analysis on the same local system
2. AI on IBM Z Sample Fraud Detection Dashboard is configured to invoke MLz AI model
3. Add items to cart



1. AI model training.

2. AI model deployment

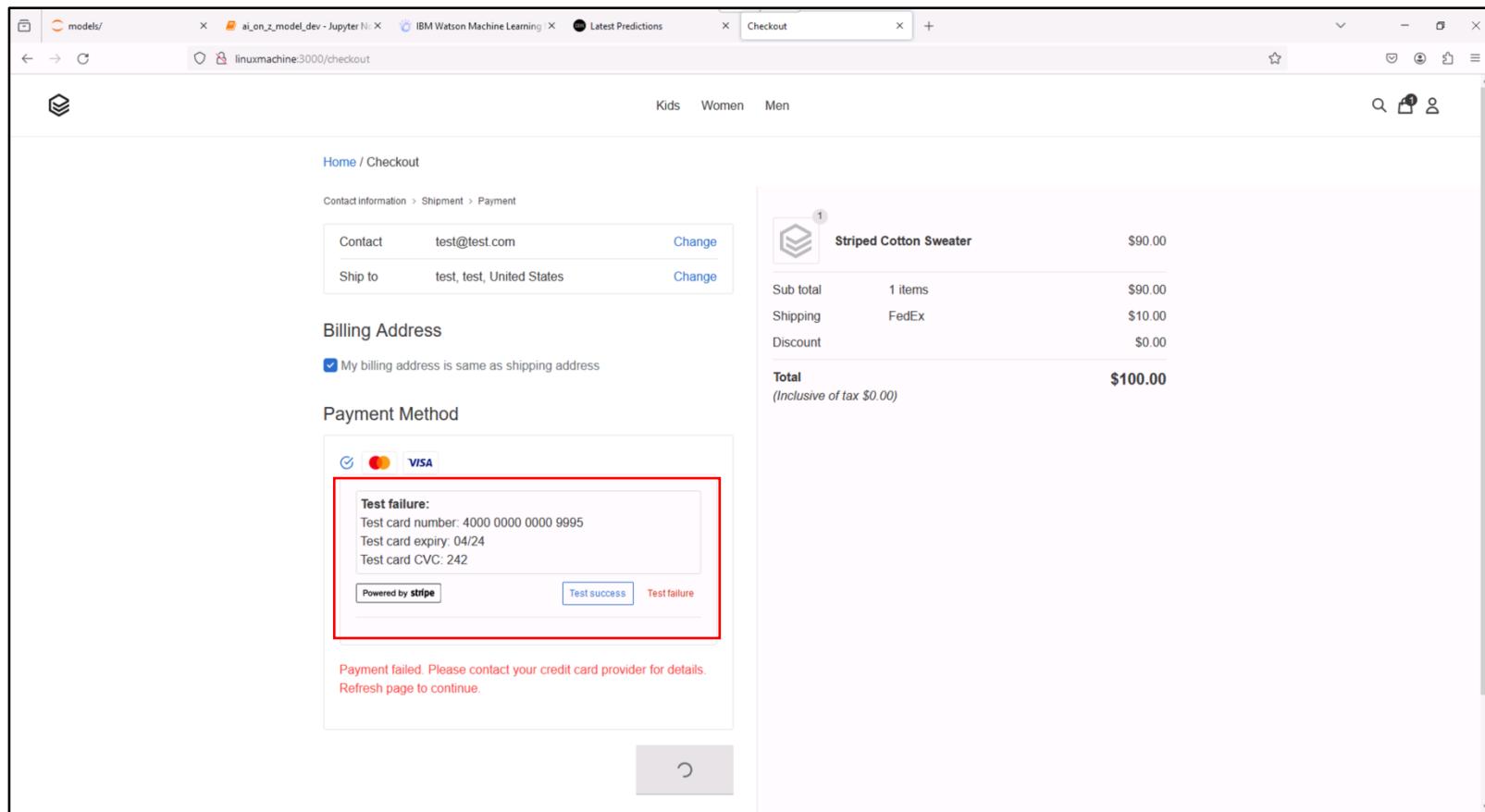
3. AI model analysis

4. AI model integration

4. Place order

- Choose test failure as payment method for fraud transaction example

Choose test success as payment method for non-fraud transaction example



1. AI model training.

2. AI model deployment

3. AI model analysis

4. AI model integration

AI model integration complete