AI on Linux on 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 Linux on Z.

Table of contents

Step 1

AI model training

We will build a fraud detection AI model by training with the provided rapid AI on Linux on 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 TIS.

All sample code for this section is within

`aionz-st-fraud-detection-tis/zST-model-training-jupyter`

Prerequisites

- Must have Python (3.9 or 3.10) installed

Dataset guidance

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

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

There are several files included within the download. You can use credit_card_transactions-ibm_v2.csv for training. 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.

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

Access rapid AI on Linux on Z development environment

1. Access sample code

   ```
   cd zST-model-training-jupyter
   ```

2. Create and activate Python virtual environment

   ```
   python -m venv env source env/bin/activate
   ```

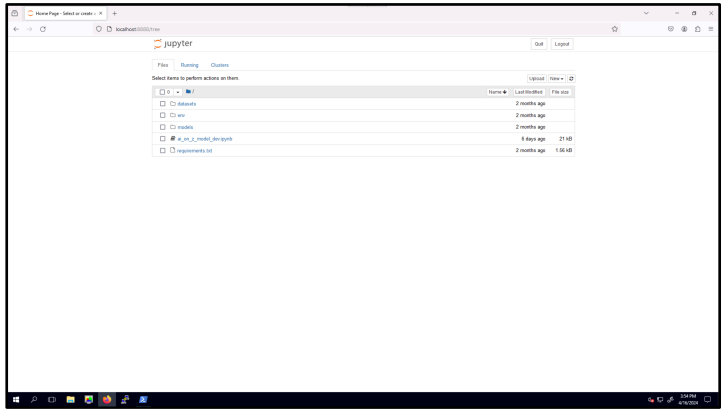
3. Install required Python packages

   ```
   pip install -r requirements.txt
   ```

4. Run Jupyter

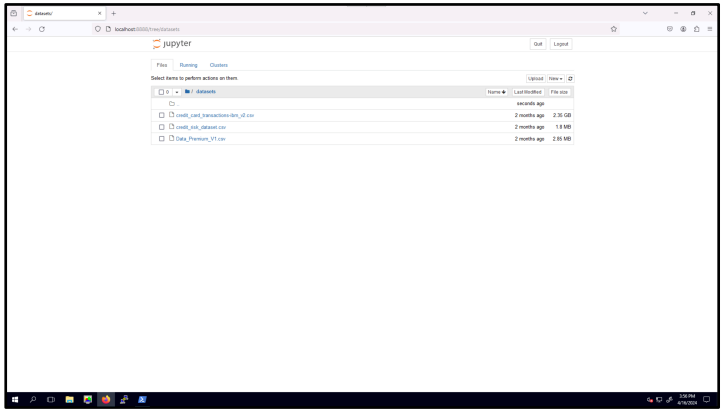
   ```
   jupyter notebook
   ```

5. View Jupyter interface
   a. Go to localhost:8888 in a web browser



Provide data

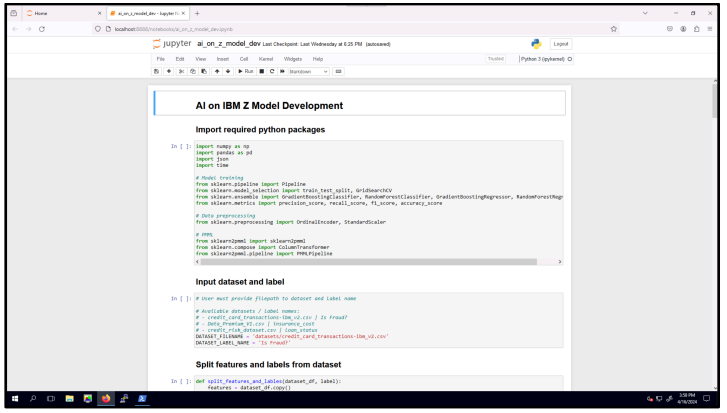
1. Your input dataset (csv) in `datasets/` directory
2. Add input data to Jupyter notebook (ai_on_z_model_dev.ipynb)
    a. Set `DATASET_FILENAME` to the path to your dataset
    b. Set `DATASET_LABEL_NAME` to the name of the column you're predicting from the dataset



Model training

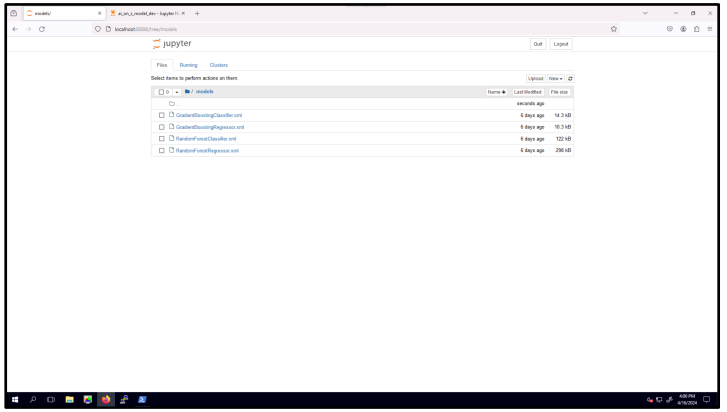
1. Step through and run all cells within Jupyter notebook (ai_on_z_model_dev.ipynb) within web browser

    Note: This may take several minutes



Access trained AI model

1. 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

Prerequisites

- Must have Docker or Podman installed

Step 2

AI model deployment

We will deploy our fraud detection AI model using TIS. We can utilize the AI Toolkit to leverage TIS for model deployment. This deployed AI model can then be integrated into applications within the Linux on Z environment.

All sample code for this section is within

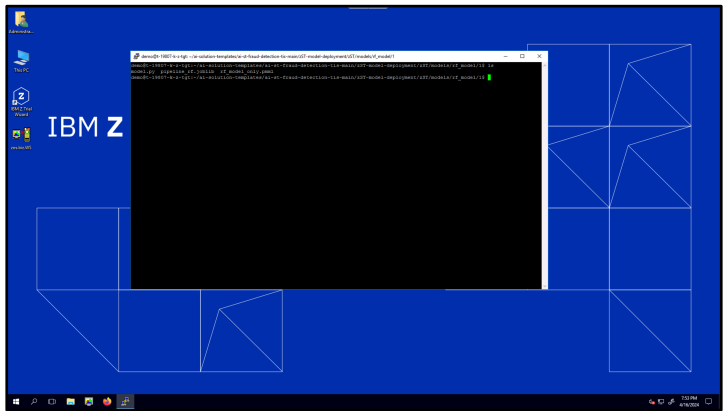
`aionz-st-fraud-detection-tis/zST-model-deployment`

Build Triton Inference Server

1. Build podman image

   ```
   podman build -t zst-tis .
   ```

Integrate AI model into Triton Inference Server



1. Add your model (.pmml file) to

   ```
   aionz-st-fraud-detection-tis/zST-model-deployment/zST/models/rf_model/1
   ```

   directory

2. Add your preprocessing .joblib file to

   ```
   aionz-st-fraud-detection-tis/zST-model-deployment/zST/models/rf_model/1
   ```

   directory

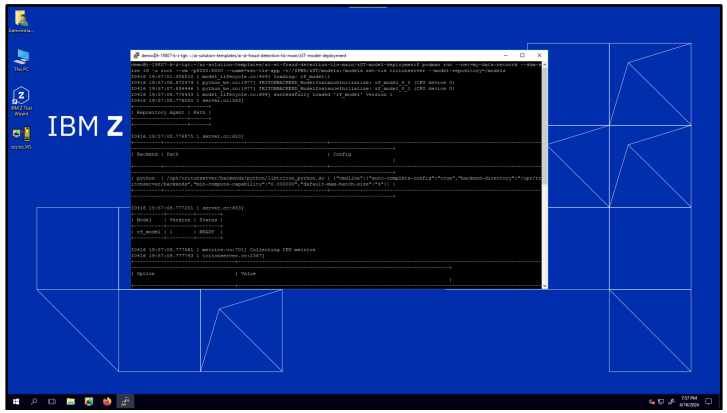
Deploy Triton Inference Server

1. Create docker network

```
podman network create my-data-network
```

2. 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
```

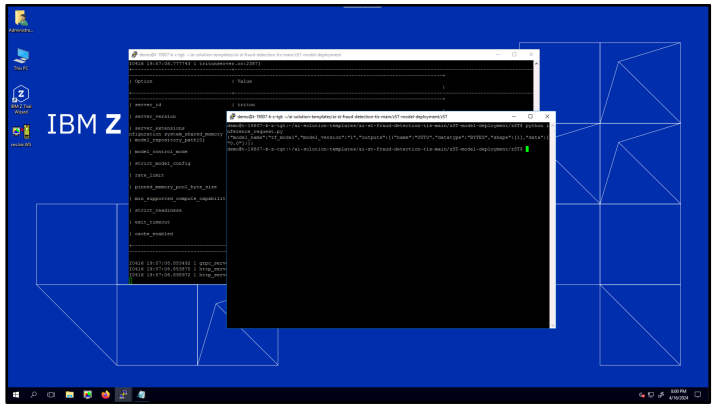


Run sample test

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

```
cd aionz-st-fraud-detection-tis/zST-model-
deployment/zST
```

```
python inference_request.py
```



☑ AI model deployment complete

Prerequisites

- Must have TIS installed
- Must have Python installed
- Must have Git installed
- Must have Docker or Podman installed

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

`aionz-st-fraud-detection-tis/zST-model-analysis`

Configure sample application

1. Set the environment variables within

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

   TIS_ENDPOINT (scoring endpoint for deployed AI model)

Build sample application

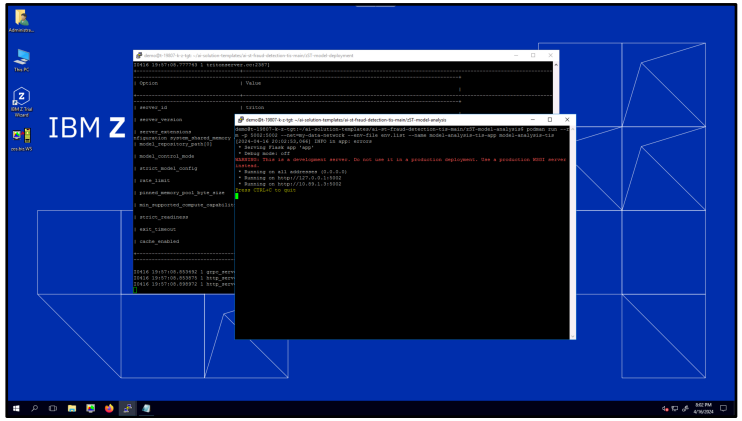
1. Run command in terminal

   ```
   podman build -t model-analysis .
   ```

Deploy sample application

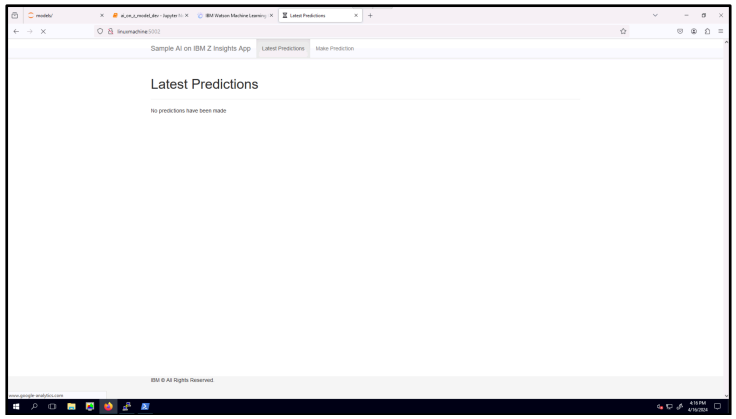
1. Run command in terminal (e.g. port 5002)

```
podman run --rm -p 5002:5002 --env-file
env.list --name model-analysis-app
model-analysis
```



Access sample application

1. View the following URL in a web browser
   http://{IP address}:{port}
   a. IP address: IP where app is deployed
   b. Port: port you used with podman run

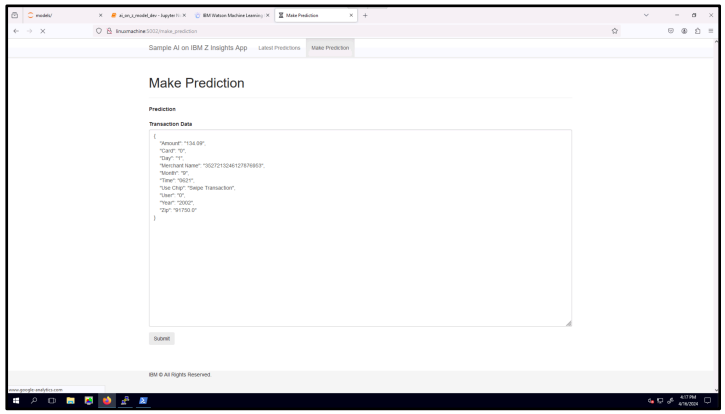


Analyze credit card events

1. Go to latest predictions tab

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"}`
3. Click submit



☑ AI model analysis complete

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.

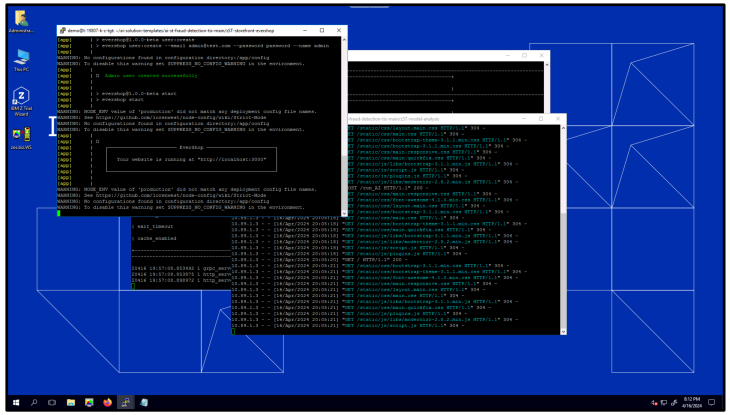
All sample code for this section is within

`aionz-st-fraud-detection-tis/zST-storefront-evershop`

Prerequisites

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

Install and deploy sample ecommerce application



1. Set app_url_w_port variable in CheckoutForm.jsx to your server IP and port (ip:port)

   ```
   ai-st-fraud-detection-tis/zST-storefronteversh
   op/packages/evershop/src/components/frontStor
   e/stripe/checkout/CheckoutForm.jsx
   ```

2. Run command in terminal (password: demo)

   ```
   podman-compose up
   ```

Configure sample ecommerce application

Access admin panel from web browser

1. Enter URL in web browser using app url (e.g. localhost)

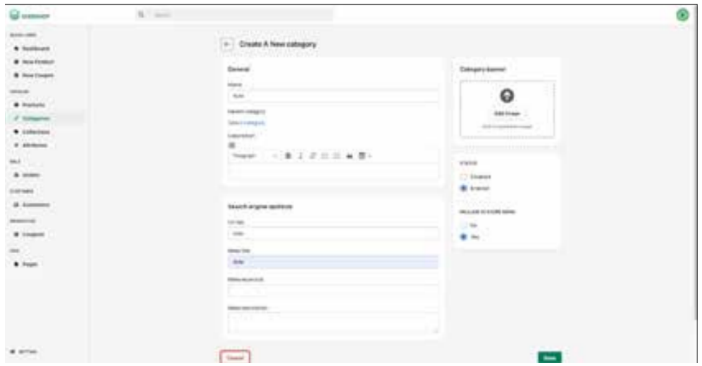
   http://localhost:3000/admin

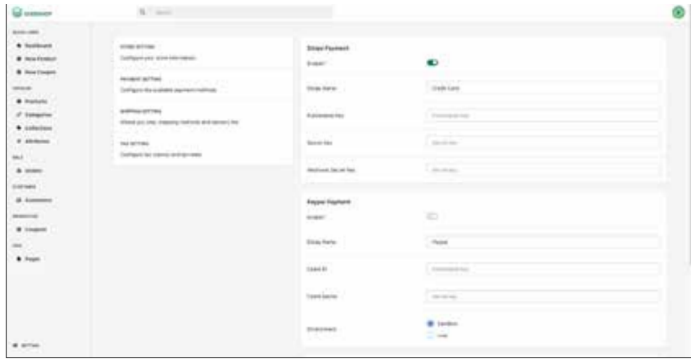
2. Login with default admin credentials

   Email: admin@test.com
   Password: password

Add products

1. Create categories
    a. Click categories from the catalog section
    b. Click new category
    c. Add category details
        i. Add name (e.g. Kids)
        ii. Add URL key
        iii. Add meta title
        iv. Change status to enable
        v. Change include in store menu to yes
    d. Click save
2. Add products
    a. Click products from the catalog section
    b. Click new product
    c. Add category details
        i. Make sure to change add category
        ii. Make sure to change status to enable
        iii. Make sure to change visibility to visible





Configure store settings

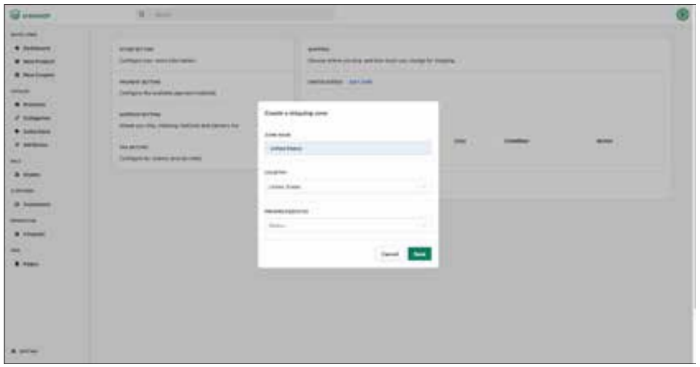
1. Enter URL in web browser using app URL (e.g. localhost)
2. Click settings on the bottom left
3. Click store setting

Configure payment settings

1. Click payment setting
2. Enable stripe payment
3. Click save

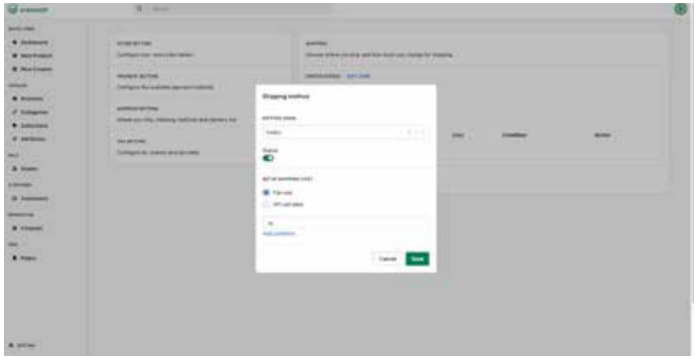
Configure shipping settings

1. Click shipping setting
2. Add shipping zone
    a. Click create new shipping zone
    b. Add shipping details
3. Click save



Add payment methods

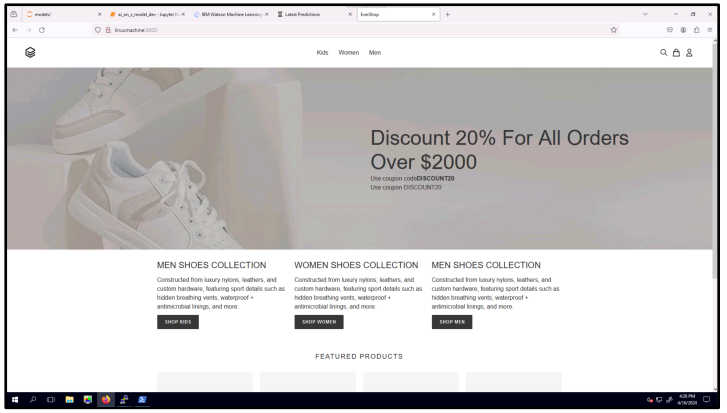
1. Click add method
2. Add method name (e.g. FedEx)
3. Enable status
4. Add flat rate (e.g. 10)
5. Click save



Access sample ecommerce application

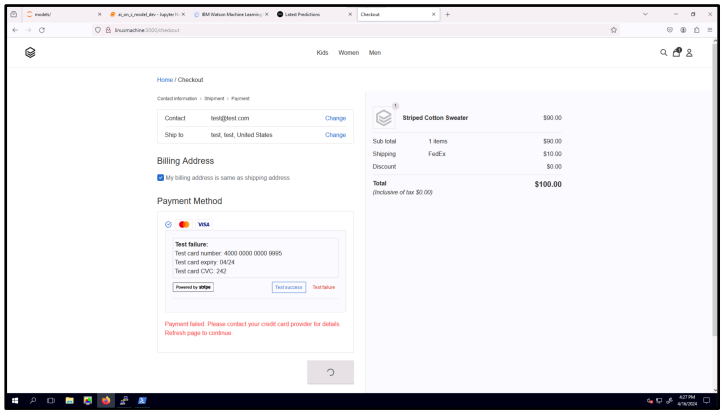
1. Enter URL in web browser using app URL (e.g. localhost)

   http://localhost:3000



Use fraud detection AI model with EverShop Storefront

1. Make sure you have AI on Linux on Z Sample Fraud Detection Dashboard deployed for inferencing and analysis on the same local system
2. AI on Linux on Z Sample Fraud Detection Dashboard is configured to invoke TIS AI model
3. Add items to cart
4. Place order
   a. Choose test failure as payment method for fraud transaction example
      Choose test success as payment method for non-fraud transaction example



☑ AI model integration complete