# Configuring custom machine learning provider scoring endpoint with IBM Watson OpenScale

There are two ways to configure Custom Machine Learning provider with Watson OpenScale:

1. To provide a discovery url, using which one can list down the available endpoints and the select the endpoint to configure it with OpenScale.
2. To configure the individual scoring endpoint.

This document focuses on the point b.

## Steps to configure Custom machine learning provider scoring endpoint

1. Open IBM Watson OpenScale console
2. Click on the “Add machine learning” provider link and provide a suitable name to the ML provider. And then click on the Edit icon under “Connection”.

A screenshot of a cell phone

Description automatically generated

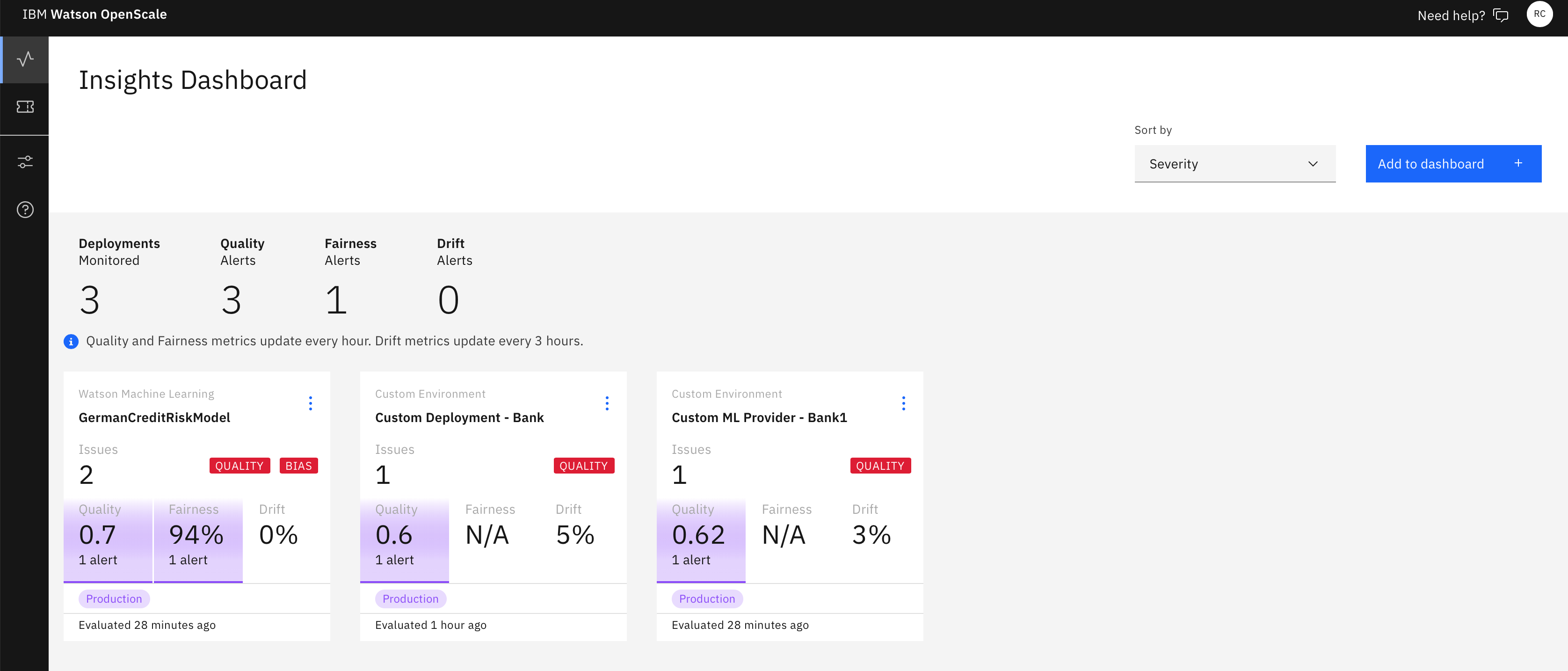
1. Under the Service provider type select “Custom Environment” and provide Username and Password to your Custom Environment.

Note: Make sure to de-select the “Use an API to get a list of models”.

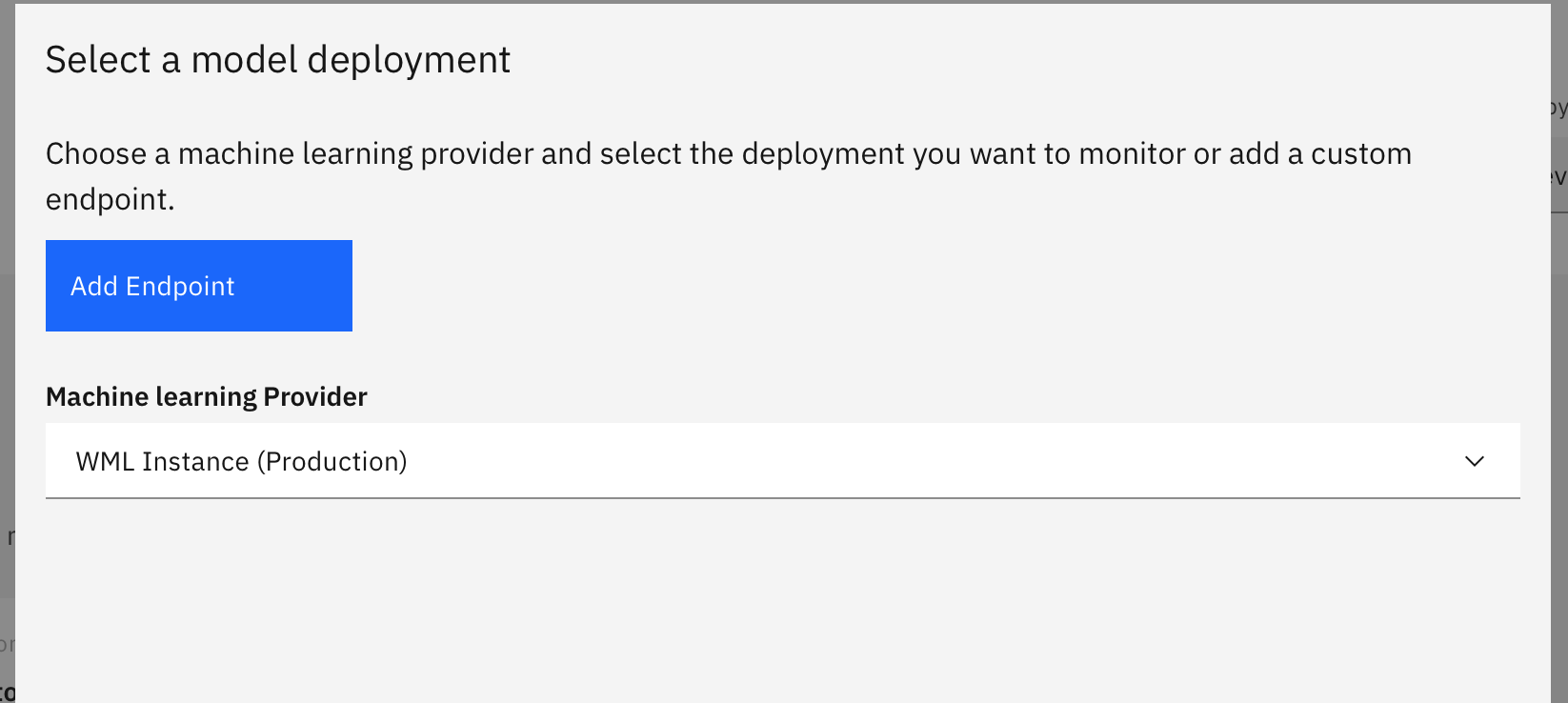
A screenshot of a cell phone

Description automatically generated

1. Based on the model that you are using for validation, whether it is for “Pre-production” usage or “Production” usage, select the option accordingly. For the purpose of this document, we are selecting “Production” type. And then Click on “Save”
2. Doing so, this would save the Custom environment with OpenScale.
3. Click on the “Insights dashboard” icon on the left nav bar, which would open up the Insights dashboard.



1. Click on the “Add to dashboard” link which is on the right side of the page, which would pop up a window listing down the ML providers.



1. Click on “Add Endpoint”, which is where we would be adding the Custom Engine scoring endpoint.
2. In the successive page, select the Custom ML Engine provider that we specified in step 2. Followed by specifying the Deployment Name and Scoring Endpoint link. Then click on “Save”

A screenshot of a social media post

Description automatically generated

1. You would get a confirmation dialog like below, where you can click on “Configure monitors”

A screenshot of a cell phone

Description automatically generated

1. From here on, based on the model type that is being used, configure the model accordingly. For demonstration purpose, the model that is being used for this document is a binary classification against structured numerical/categorical data.

A screenshot of a cell phone

Description automatically generated

1. Specify the training data that is used to train the model

A screenshot of a social media post

Description automatically generated

1. Select the scoring method to establish the input/output schema for the model

A screenshot of a cell phone

Description automatically generated

1. In the successive screens, select the label column, features that are used to train the model, prediction and probability columns.
2. Configure Drift monitor
   1. Using the <https://github.com/IBM-Watson/aios-data-distribution/blob/master/training_statistics_notebook.ipynb> to generate the drift detection model.
   2. Upload the model and then specify the drift threshold and also the min records that are to be used for detection.
3. Configure other monitors like Quality and Fairness as needed.
4. Perform the continuous scoring for the model and thereby add the records to the payload logging table using the scoring\_paylaod API. You can as well use the Python SDK to add the scoring payloads using the API `subscription.payload\_logging.store(records=records)`.
5. Run the monitors using MRM dashboard to trigger the evaluation of the configured monitors.

A screenshot of a cell phone

Description automatically generated