NIM Metrics: Visualization

Related projects

- opendatahub-io/odh-model-controller
- opendatahub-io/odh-dashboard

Prerequisites

- oc
- jq
- curl
- RHOAI instance running the development image of odh-model-controller (see PR)
- A running NIM serving

Currently we use AI-Dev04. **Make sure you're logged in from your terminal using oc.**

Assumptions

The following serving metadata is assumed, if your environment differs, make sure to update the various commands as you go.

- Data Science project name: *tomer-test-2*
- Serving name: *nim-deploy*
- Model served: *llama3-8b-instruct*

Verify backend graph objects

We currently have 10 graph types, 4 pre-existing and 6 new. For every Data Science project, the backend creates a ConfigMap encapsulating the graph objects suitable for the running serving.

In our case, we should expect the following existing graph types: REQUEST_COUNT, MEAN_LATENCY, CPU_USAGE, and MEMORY_USAGE. As well as the following new graph types: KV_CACHE, CURRENT_REQUESTS, TOKENS_COUNT, TIME_TO_FIRST_TOKEN, TIME_PER_OUTPUT_TOKEN, and REQUEST_OUTCOMES.

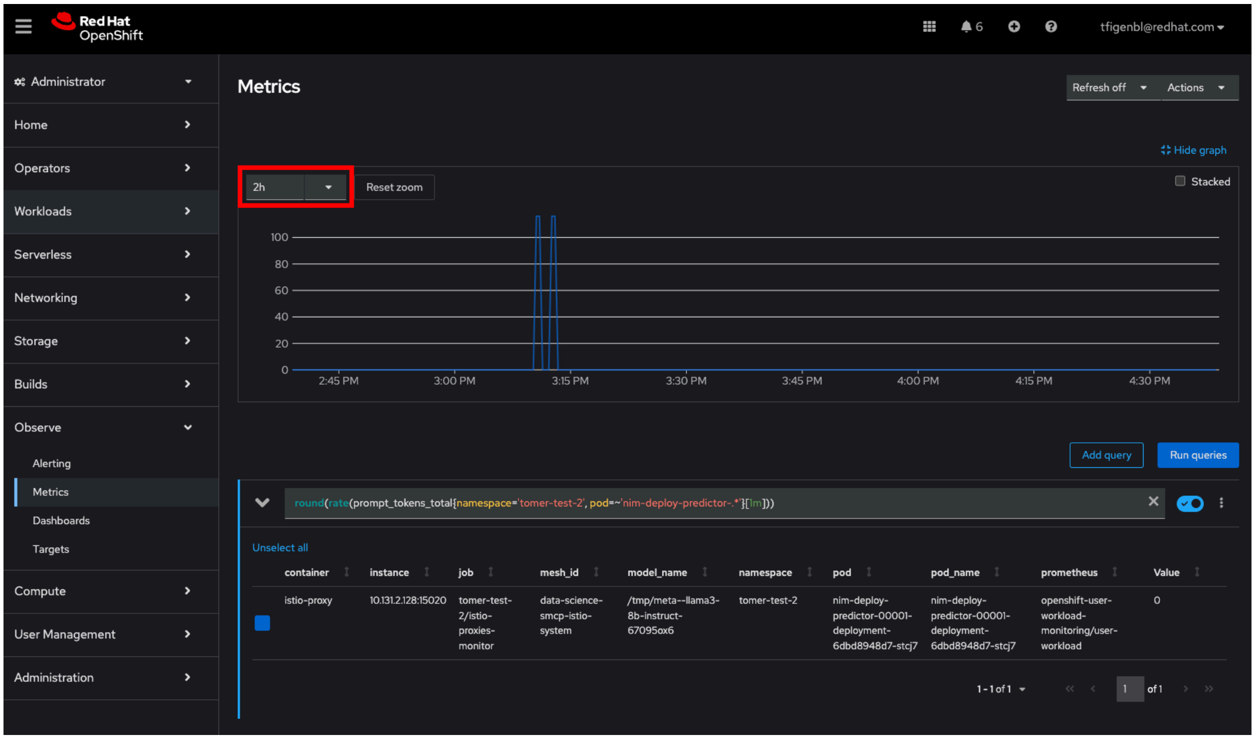
Use the following command to get the object implementation and note the queries encapsulated within it, change the type to check another graph object:

```
oc get configmap -n tomer-test-2 nim-deploy-metrics-dashboard \
-o jsonpath='{.data.metrics}' | \
jq '.config[] | select(.type=="TOKENS_COUNT")'
```

```
{
  "title": "Tokens count",
  "type": "TOKENS_COUNT",
  "queries": [
    {
      "title": "Total prompts token",
      "query": "round(rate(prompt_tokens_total{namespace='tomer-test-2',
pod=~'nim-deploy-predictor-.*'}[1m]))"
    },
    {
      "title": "Total generation token",
      "query":
"round(rate(generation_tokens_total{namespace='tomer-test-2',
pod=~'nim-deploy-predictor-.*'}[1m]))"
    }
  ]
}
```

Verify queries data

If you need to verify the queries have current data, in OpenShift’s dashboard, go into Observe -> Metrics, and run the query in question. Make sure to update the graph time frame based on traffic, for some metrics:



```
round(rate(prompt_tokens_total{namespace='tomer-test-2', pod=~'nim-deploy-predictor-.*'}[1m]))
```

container	instance	job	mesh_id	model_name	namespace	pod	pod_name	prometheus	Value
istio-proxy	10.131.2.128:15020	tomer-test-2/istio-proxies-monitor	data-science-smcp-istio-system	/tmp/meta--llama3-8b-instruct-67095ox6	tomer-test-2	nim-deploy-predictor-00001-deployment-6dbd8948d7-stcj7	nim-deploy-predictor-00001-deployment-6dbd8948d7-stcj7	openshift-user-workload-monitoring/user-workload	0

Enable NIM metrics

Currently, NIM metrics are disabled in the RHOAI dashboard; it will be enabled as part of the ongoing work effort. See NVPE-18. To enable it locally on your station. Save the following diff file, and apply it:

enable_nim_graphs.diff

```
git apply enable_nim_graphs.diff
```

Create serving traffic

From a separate terminal, forward traffic from your local station port 4321 to the serving port 80. If port 4321 is occupied on your station, you can select a different port, just remember to update the commands accordingly:

```
oc port-forward -n tomer-test-2 svc/nim-deploy-predictor-00001-private
4321:80
```

Once the traffic is forwarded, you can run commands against your local station, and the traffic will be forwarded to the cluster.

Get available models

```
curl -s http://localhost:4321/v1/models | jq
```

Chat with the model

```
curl -H "Content-Type: application/json"
http://localhost:4321/v1/chat/completions -sd \
'{
  "model": "meta/llama3-8b-instruct",
  "messages": [
    {"role":"user","content":"What is Red Hat OpenShift AI?"},
    {"role": "user", "content": "What is NVIDIA NIM?"}
  ],
  "temperature": 0.5,
  "top_p": 1,
  "max_tokens": 1024,
  "stream": false
}' | jq
```

Create load

Use the following command to send multiple parallel chat requests to the model, this will generate load that will be reflected in the metrics data. The following command will generate 120 requests, change this value to fit your needs:

```
for i in {1..120}; do curl -H "Content-Type: application/json"
http://localhost:4321/v1/chat/completions -sd \
'{
  "model": "meta/llama3-8b-instruct",
  "messages": [
    {"role":"user","content":"What is Red Hat OpenShift AI?"},
    {"role": "user", "content": "What is NVIDIA NIM?"}
  ],
  "temperature": 0.5,
  "top_p": 1,
  "max_tokens": 1024,
  "stream": false
}' 2>&1 > /dev/null &; done
```

Connect from local frontend

You can run odh-dashboard against a remote backend, i.e. the RHOAI instance you’re connected to with the development version of odh-model-controller (AI-Dev04). Run the following command to create a local server against the remote cluster:

```
(npm run build && cd frontend && OC_PROJECT=redhat-ods-applications
ODH_APP=rhods-dashboard npm run start:dev:ext)
```

Go to the nim-deploy inside the project *tomer-test-2,* press the NIM tab. There you should see the graphs.