Dynamic Ecommerce Discounts with Redpanda

Git commit with doc: 0664b37

HTML version

1. Introduction

This project is a companion to the Example Next.js Ecommerce Store for Snowplow.

It allows you to test this demo locally, using LocalStack, and in the AWS cloud.

Its Architecture is designed so a developer can quickly and easily set up these two environments and test the project.

2. Architecture

- The **ecommerce-nextjs-example-store** is a Next.js application that generates tracking events.
- The **stream-collector** component sends these events via Kinesis to the **[snowbridge]** component.
- The **snowbridge** component enriches these events, inserts more information (via **[enrich]** component), and sends them to Redpanda.
    - Read more about the **enrich** component here: https://docs.snowplow.io/docs/pipeline-components-and-applications/enrichment-components/enrich-kinesis/.
    - Read more about the **snowbridge** component here: https://docs.snowplow.io/docs/destinations/forwarding-events/snowbridge/.

Sequence Diagram for the Architecture:

TODO

All components in this Architecture run as Docker containers via `docker compose`:

- The Snowplow's components (**[stream-collector]**, **[enrich]**, and **[snowbridge]**) are defined in the file `compose.snowplow.yaml`.
- Redpanda's infrastructure is provided by the file `compose.redpanda.yaml`.
- The apps components (**[ecommerce-nextjs-example-store]**) are defined in the file `compose.apps.yaml`.
- The infrastructure to provide the AWS resources locally (Kinesis, DyanmoDB, etc) is created by LocalStack.
    - Read the file `compose.localstack.yaml`.
- These components and resources are created in AWS using Terraform scripts.
    - There is another document, in `docs/terraform` folder, explaining the details.

3. Prerequisites

1. Start a Ubuntu Linux (it can be running on a WSL2 environment) terminal.
2. Make sure you have docker (and docker compose) installed.
3. Clone this project with Git and cd to it.
4. Create a file `docker/.env` (from `docker/.env.sample`) and configure the AWS variables on it.

> You don’t need Java or Node.js configured on your machine to follow the steps below. **You only need a Bash terminal and a Docker installation.**

4. Steps (to run this application as is)

Step 1 → Start the containers

```
$ ./docker/up.sh
```

Tips:

1. You can press `Ctrl` + `C` at any time. The docker containers will remain running.
2. If there is no file `docker/.env` in the project, this script will try to locate it in a file named `../dynamic-ecommerce-discounts-with-redpanda.env` and copy it to `docker/.env`. This allows you to call `git clean -fdX` at any time you want without losing your configuration.
    a. If the file `../dynamic-ecommerce-discounts-with-redpanda.env` does not exists, it will copy the file `docker/.env.sample` to `docker/.env` and use it.
3. You can pass "services" as an argument option to this script. It will list the options you can pass to it by adding the suffix "-services":

    ```
    $ ./docker/up.sh services
    apps
    localstack
    redpanda
    snowplow
    ```

4. By adding the "-services" to one of the options listed above, you will start only the services listed in the file `copose.<service>.yaml`. So, this will start only the kafka services (services listed in `compose.kafka.yaml`):

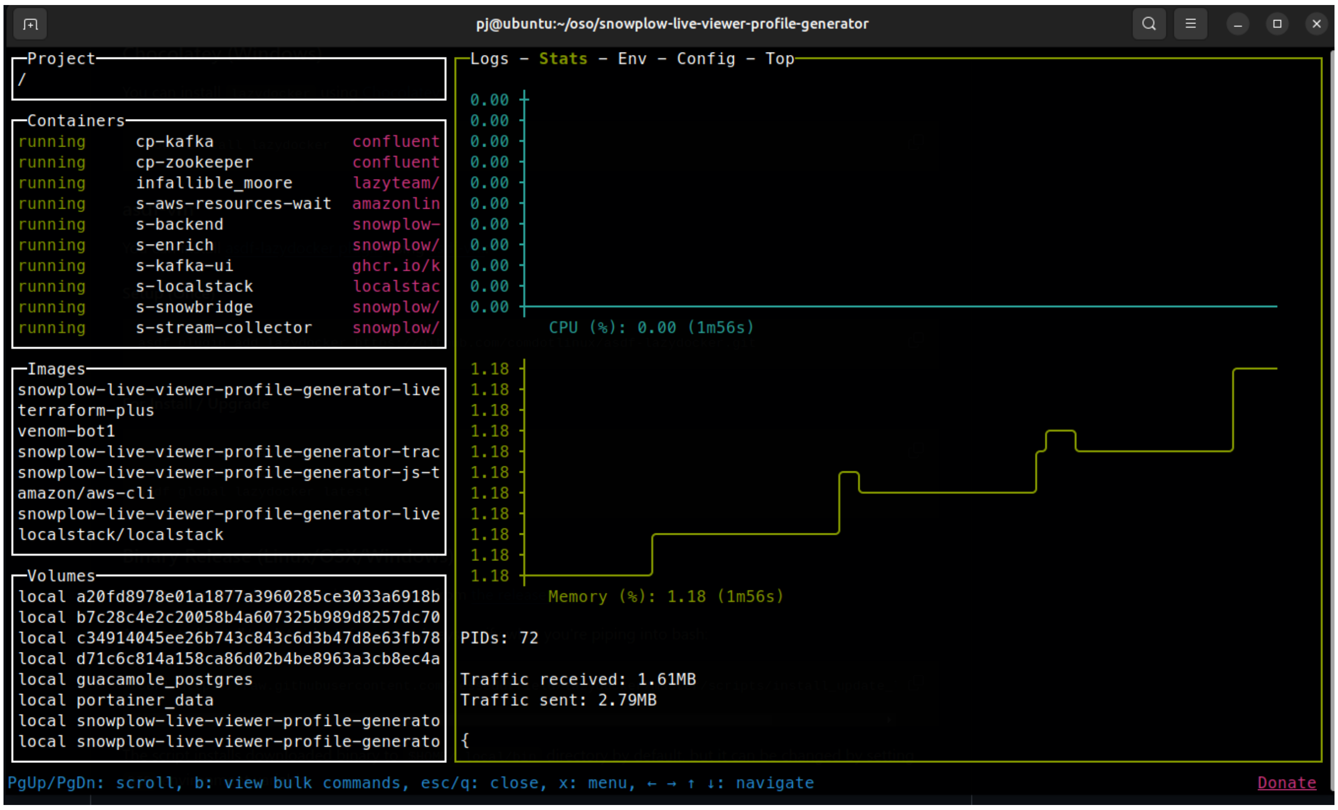
    ```
    $ ./docker/up.sh snowplow-services
    ```

Step 2 → Know the URL provided by the services

1. **LocalStack**: https://app.localstack.cloud/
2. **Redpanda**: http://localhost:8080
    a. User / password: jane / some-other-secret-password

Step X → (optional) Use LazyDocker to monitor the containers and logs

```
$ ./docker/lazy.sh
```



```
pj@ubuntu:~/oso/snowplow-live-viewer-profile-generator

┌Project──────────────────────────────────────────┐ ┌Logs - Stats - Env - Config - Top──────────────
│/                                                │ │ 0.00 ┤
└─────────────────────────────────────────────────┘ │ 0.00 ┤
┌Containers───────────────────────────────────────┐ │ 0.00 ┤
│running      cp-kafka                  confluent │ │ 0.00 ┤
│running      cp-zookeeper              confluent │ │ 0.00 ┤
│running      infallible_moore          lazyteam/ │ │ 0.00 ┤
│running      s-aws-resources-wait      amazonlin │ │ 0.00 ┤
│running      s-backend                 snowplow- │ │ 0.00 ┤
│running      s-enrich                  snowplow/ │ │ 0.00 ┤
│running      s-kafka-ui                ghcr.io/k │ │ 0.00 ┤
│running      s-localstack              localstac │ │ 0.00 ┼──────────────────────────────────────
│running      s-snowbridge              snowplow/ │ │         CPU (%): 0.00 (1m56s)
│running      s-stream-collector        snowplow/ │ │ 1.18 ┤
└─────────────────────────────────────────────────┘ │ 1.18 ┤
┌Images───────────────────────────────────────────┐ │ 1.18 ┤
│snowplow-live-viewer-profile-generator-live      │ │ 1.18 ┤
│terraform-plus                                   │ │ 1.18 ┤
│venom-bot1                                       │ │ 1.18 ┤
│snowplow-live-viewer-profile-generator-trac      │ │ 1.18 ┤
│snowplow-live-viewer-profile-generator-js-t      │ │ 1.18 ┤
│amazon/aws-cli                                   │ │ 1.18 ┤
│snowplow-live-viewer-profile-generator-live      │ │ 1.18 ┤
│localstack/localstack                            │ │ 1.18 ┼
└─────────────────────────────────────────────────┘ │         Memory (%): 1.18 (1m56s)
┌Volumes──────────────────────────────────────────┐ │
│local a20fd8978e01a1877a3960285ce3033a6918b      │ │PIDs: 72
│local b7c28c4e2c20058b4a607325b989d8257dc70      │ │
│local c34914045ee26b743c843c6d3b47d8e63fb78      │ │Traffic received: 1.61MB
│local d71c6c814a158ca86d02b4be8963a3cb8ec4a      │ │Traffic sent: 2.79MB
│local guacamole_postgres                         │ │
│local portainer_data                             │ │{
│local snowplow-live-viewer-profile-generato      │ │
│local snowplow-live-viewer-profile-generato      │ │
└─────────────────────────────────────────────────┘ └───────────────────────────────────────────────
PgUp/PgDn: scroll, b: view bulk commands, esc/q: close, x: menu, ← → ↑ ↓: navigate          Donate
```

5. Clean up steps

Step 1 → Stop the containers

To stop all the containers, type:

```
$ ./docker/down.sh
```

Step 2 → Clean up

To remove all the containers and images, type:

```
$ ./docker/clean.sh
```

Warnings:

1. The script `clean.sh` will destroy any data generated by these containers.

6. References

LocalStack

Redpanda

- Docker Compose Labs
  - Start a Single Redpanda Broker with Redpanda Console in Docker
- Redpanda Self-Managed Quickstart

7. Demo videos