**KAFKA VERTICA INTEGRATION**

**Note:**

This document outlines the process of creating a Kafka scheduler within Vertica, which I have developed using Kubernetes (K8s) as part of a containerized Vertica setup. Vertica is installed and managed within containers to optimize deployment, scalability, and resource management.

Official documentation link - [Containerized Kafka Scheduler | Vertica 24.3.x](https://docs.vertica.com/24.3.x/en/containerized/kafka/)

**Description:**

The Vertica-Apache Kafka integration includes a scheduler—a configurable mechanism that automates the process of consuming data from Kafka and loading it into a Vertica database. This scheduler enables seamless data ingestion from Kafka streams, enhancing data availability and analysis capabilities within Vertica.

**Prerequisites**

Before setting up the Kafka scheduler in Vertica, ensure the following tools and resources are available:

- Kubernetes: Version 1.21.1 or higher

- Helm: Version 3.5.0 or higher

- kubectl: Kubernetes command-line tool

- Kafka Cluster: A fully configured and accessible Kafka cluster for data streaming

**Steps to Set Up the Kafka Scheduler in Vertica**

**1. Add the Helm Charts**

To simplify deployment, Vertica provides the Kafka Scheduler as a Helm chart. First, add these charts to your local Helm repository:

$ helm repo add vertica-charts https://vertica.github.io/charts

$ helm repo update

**2. Create a Namespace for Kafka**

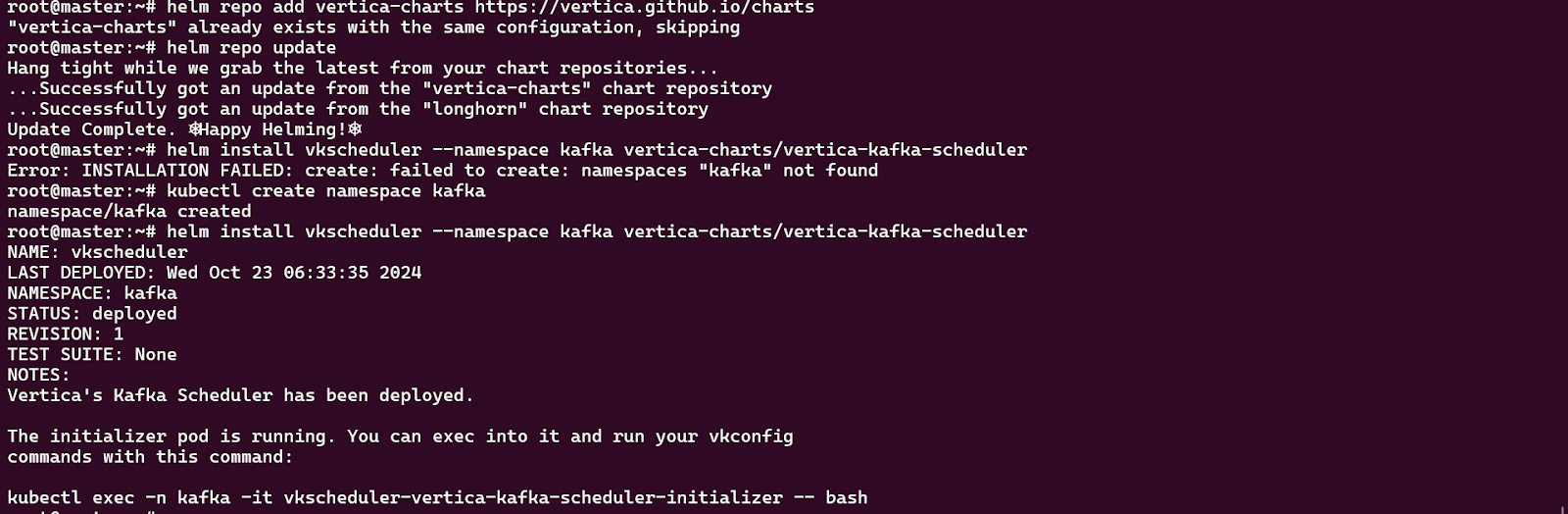
Next, create a dedicated namespace for Kafka to organize and isolate its resources within Kubernetes:

$ kubectl create namespace kafka

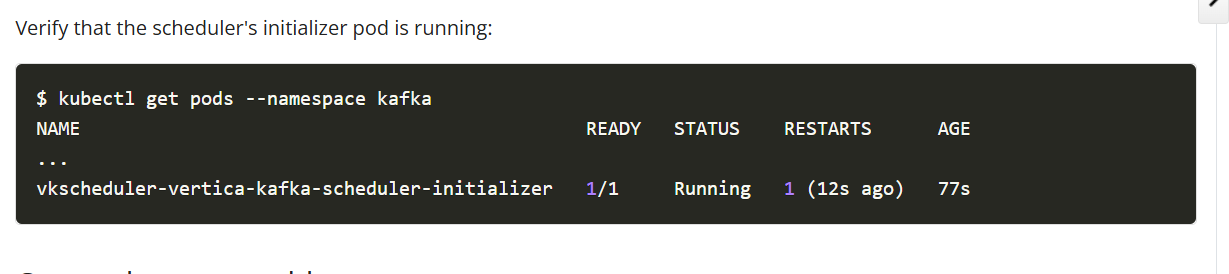
**3) Install the scheduler**

Install the scheduler Helm chart to start the scheduler in initializer mode. The following helm install command deploys a scheduler named vkscheduler in the kafka namespace:

$ helm install vkscheduler --namespace kafka vertica-charts/vertica-kafka-scheduler



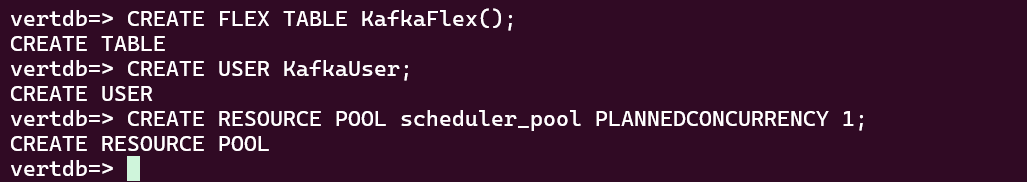
**4) verify scheduler**



**5 ) Create Target Table in vertica database**

1. **Access the Vertica pod and start the Vertica service using the following two commands.**
2. **kubectl exec -it verticadb-sc1-0 -n vertica -- /bin/bash**
3. **vsql -U dbadmin -d vertdb**

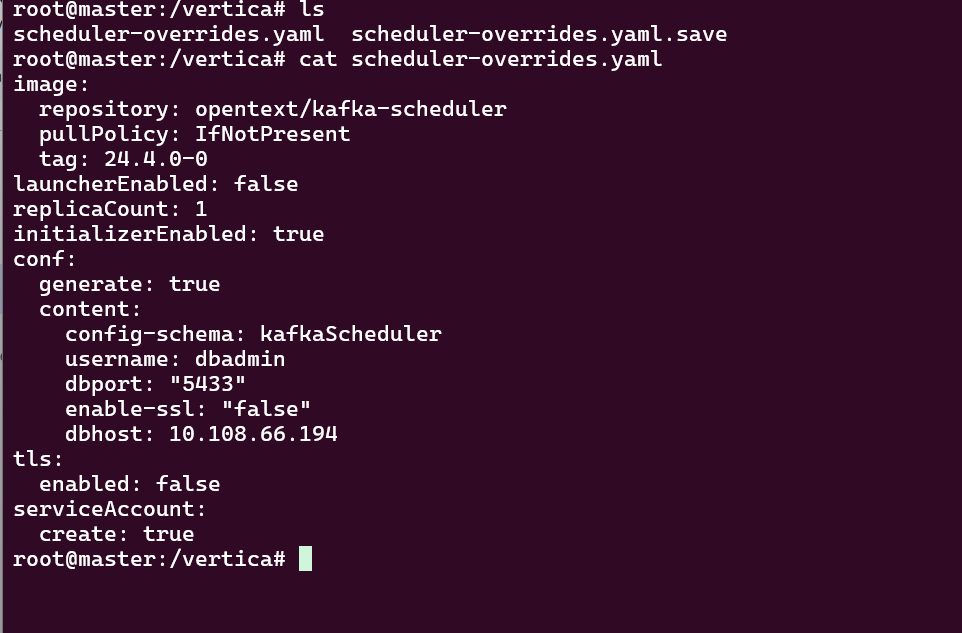
**B) Create Target table (https://docs.vertica.com/24.3.x/en/containerized/kafka/#create-the-target-table)**



**6) Override scheduler configuration(** [**https://docs.vertica.com/24.3.x/en/containerized/kafka/#override-scheduler-configuration**](https://docs.vertica.com/24.3.x/en/containerized/kafka/#override-scheduler-configuration) **)**

To ensure compatibility with our Vertica version, we need to override specific scheduler configuration settings using the Helm upgrade method.

First, create a YAML file named scheduler-overrides.yaml following the format specified in the documentation.



Once you have created the file, execute the following command:

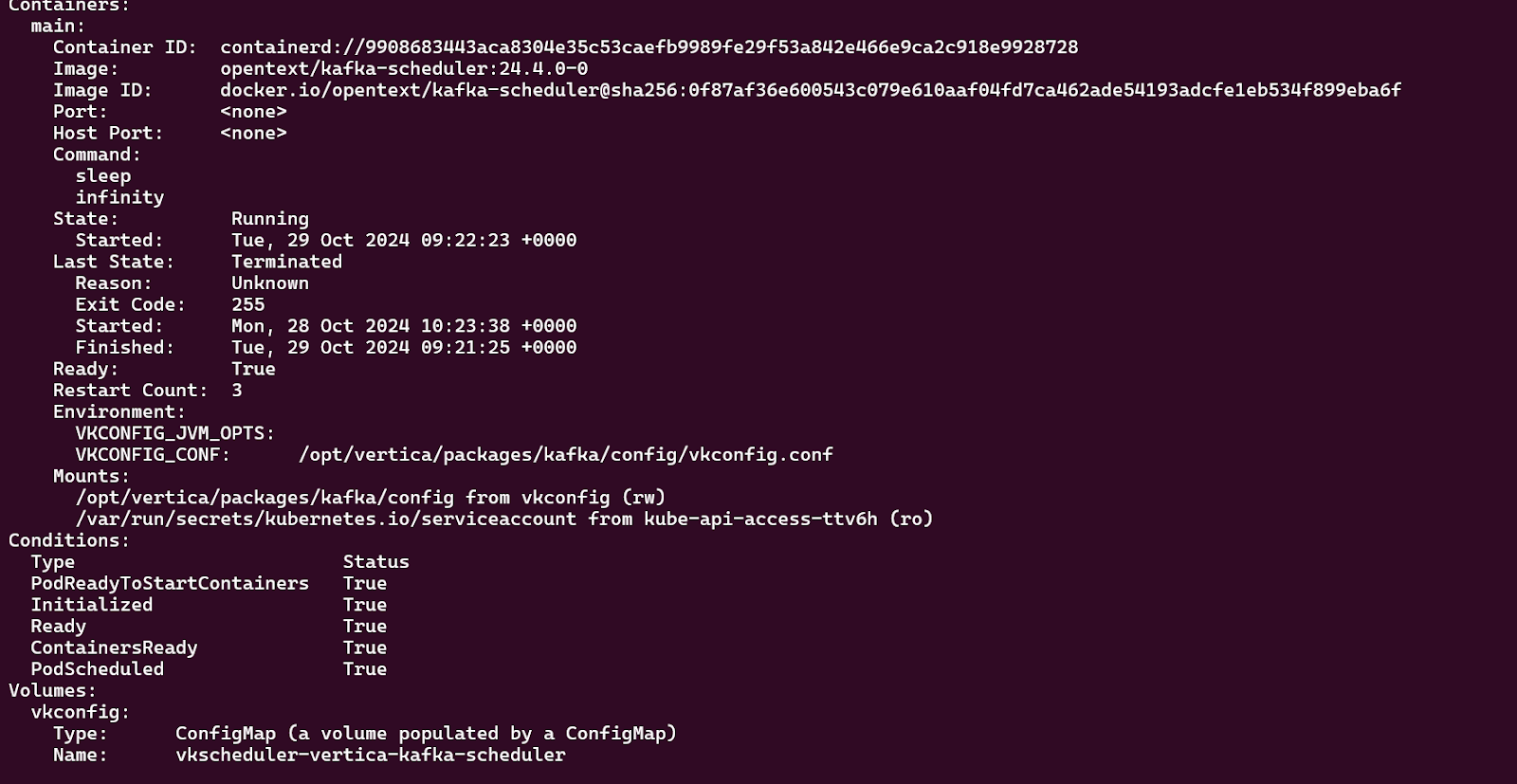
helm upgrade vkscheduler --namespace kafka vertica-charts/vertica-kafka-scheduler -f scheduler-overrides.yaml

This will apply the new scheduler configurations to the Vertica Kafka Scheduler.

7) verify the changes -

Run command:

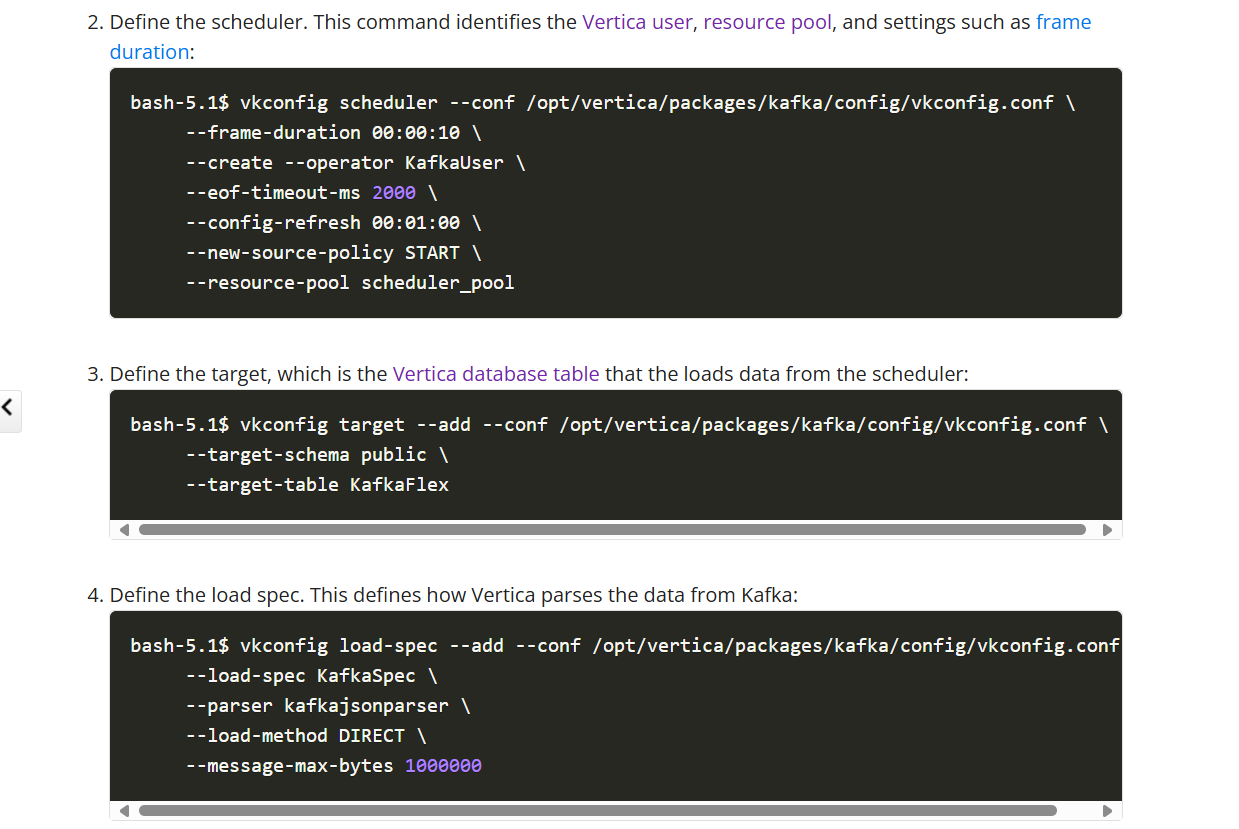
kubectl describe pod vkscheduler-vertica-kafka-scheduler-initializer -n kafka

And verify image tag - 

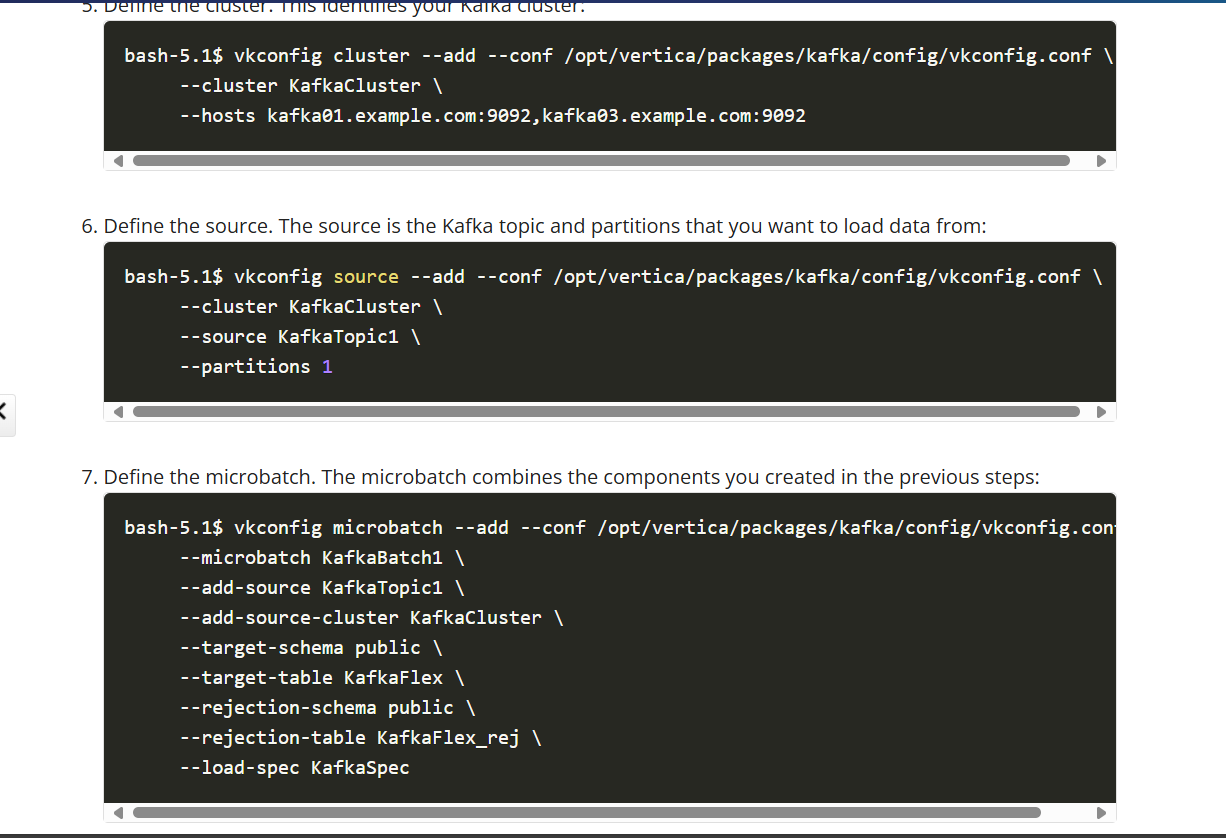
8) Configure the scheduler -

<https://docs.vertica.com/24.3.x/en/containerized/kafka/#configure-the-scheduler>

Follow the steps above. Sometimes, when we add a configuration, it may not respond. In such cases, press Ctrl+C and add another conf

first 3 should be added exactly 

In the 5th step, add the Kafka broker host and port (from Ambari). In the 6th step, specify the source (Kafka topic) and see that the partition is 1, as we cannot set the partition to 0 in this configuration so Create a Kafka topic with two partitions (partition 0 and partition 1) and send data to the second partition (partition 1). Finally, in the 7th step, use the --add-source option to add the Kafka topic.

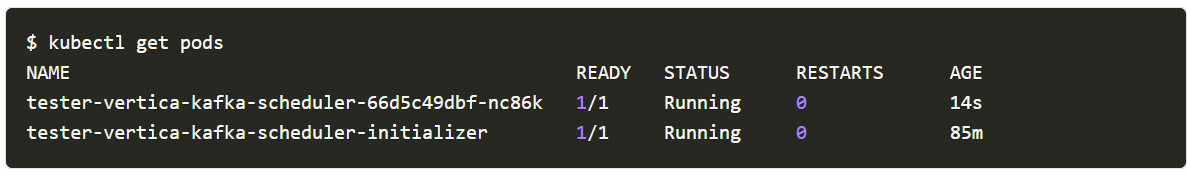


9) To launch the scheduler, make sure to use the parameter --reuse-values to prevent it from using default values. Use the following command:

helm upgrade --namespace kafka vkscheduler vertica-charts/vertica-kafka-scheduler \

--set "launcherEnabled=true" \

--reuse-values

Check status - 

If the launcher failed check the logs for launcher

10. Test your scheduler by sending data to the producer and verify the send data from the flex table

**Note:** Avoid using SELECT \* FROM flex\_table. Instead, select the specific key you sent to the flex table.