**Lab : The Complete CLI Workflow (File Ingest & Real-time Stream)**

**Goal:** Use the Redpanda command-line tools (rpk and rpk connect) to manage a Redpanda Cloud topic and build a complete data pipeline that first ingests data from a file and then streams it back to your console in real-time.

# **Purpose of the Lab**

This lab provides a comprehensive, end-to-end experience using only Redpanda's command-line tools. It is designed for operators, SREs, and developers who prefer a configuration-driven, no-code approach to data streaming. By completing this lab, you will master the full lifecycle of a simple pipeline: creating cloud resources, securely managing credentials, building a batch ingestion connector, and then building a separate real-time streaming connector to observe the data flow.

# **Prerequisites**

* A Redpanda Cloud account with a running cluster.
* The rpk CLI and rpk connect binary installed on your local machine.

# **Project Layout**

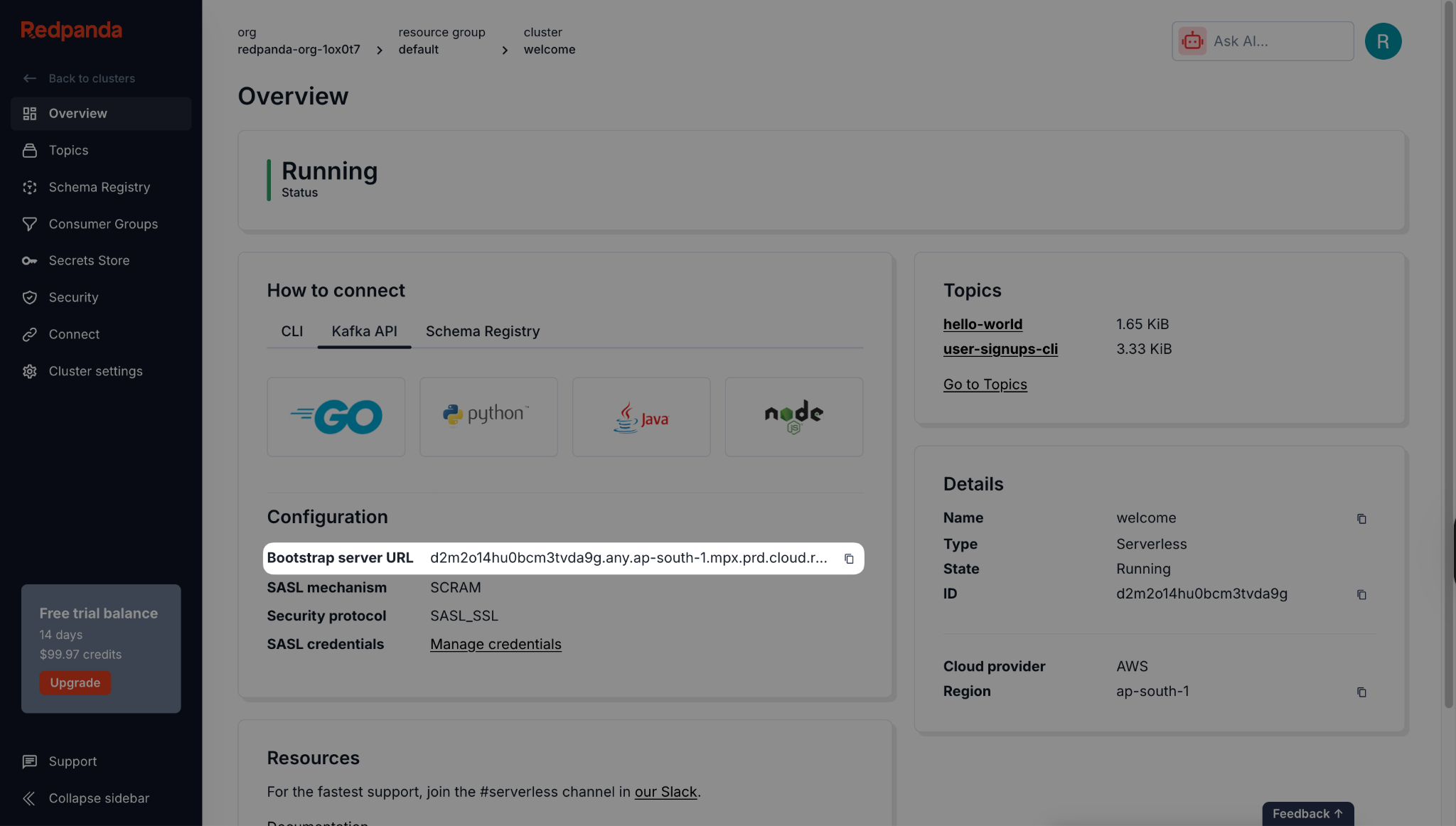
You will create a single directory for this lab which will contain all the necessary files.

|  |
| --- |
| rp-cli-cloud-lab/ ├── users.jsonl ├── file-to-cloud.yaml ├── cloud-to-stdout.yaml └── .env |

# **Part 1: Ingesting Data from a File**

## Step 1: Get Cloud Credentials and Create a User

1. **Navigate to Security:** In the [Redpanda Cloud UI](https://cloud.redpanda.com/), click on the **Security** tab in the left-hand menu.
2. **Create a User:** Go to the **Users** sub-tab. Click **Create user**. Give the user a name (e.g., cli-user) and save the generated **username** and **password** in a secure, temporary location.
3. **Go to ACLs:** Click on the **ACLs** sub-tab.
4. **Grant Permissions:** Click **Create ACL**. Select the cli-user you just created. For the lab, select the **Allow All** preset for permissions, then click **Create**.
5. **Get Broker Address:** Go to the **Clusters** page, select your cluster. On the **Overview** page, click the **Kafka API** tab and copy the **Brokers** address



## Step 2: Create an rpk Profile

1. **Log in to Redpanda Cloud:**

|  |
| --- |
| rpk cloud login |

1. **Create a Profile:** This saves your cloud connection details under a memorable name.

|  |
| --- |
| rpk profile create --from-cloud |

Follow the prompts. A common default profile name is rpk-cloud.

## 

## **Step 3: Prepare the Project**

1. **Create the project directory:**

|  |
| --- |
| mkdir rp-cli-cloud-lab cd rp-cli-cloud-lab |

1. **Create a sample data file named users.jsonl:**

**users.jsonl**

|  |
| --- |
| {"id":1,"email":"alice@example.com","country":"USA"} {"id":2,"email":"bob@example.com","country":"CA"} {"id":3,"email":"[carol@example.com](mailto:carol@example.com)","country":"UK"} |

1. **Create the destination topic on Redpanda Cloud:**

|  |
| --- |
| rpk topic create user-signups-cli --profile rpk-cloud |

1. **Create the Environment File (.env):** This file will securely hold your secrets.

|  |
| --- |
| # .env REDPANDA\_BROKERS="<YOUR\_BROKERS\_URL>" REDPANDA\_USER="<YOUR\_USERNAME>" REDPANDA\_PASS="<YOUR\_PASSWORD>" |

Replace the placeholders with the credentials for the cli-user you created in Step 1.

## **Step 4: Create the Ingestion Pipeline Configuration**

Create a file named file-to-cloud.yaml. This connector will read your local file and send it to the cloud topic.

**file-to-cloud.yaml**

|  |
| --- |
| # file-to-cloud.yaml input:  file:  paths: ["./users.jsonl"]  codec: "lines" pipeline:  processors:  - bloblang: 'root = this.merge({"ingest\_time": now().string()})' output:  kafka:  addresses: [ "${REDPANDA\_BROKERS}" ]  topic: "user-signups-cli"  tls: { "enabled": true }  sasl:  mechanism: "SCRAM-SHA-256"  user: "${REDPANDA\_USER}"  password: "${REDPANDA\_PASS}" |

## **Step 5: Run the Ingestion Pipeline**

Execute the connector using the --env-file flag to load your credentials.

rpk connect run --env-file .env ./file-to-cloud.yaml

**Expected output:** The connector will start, connect, process the three records, and shut down.

## 

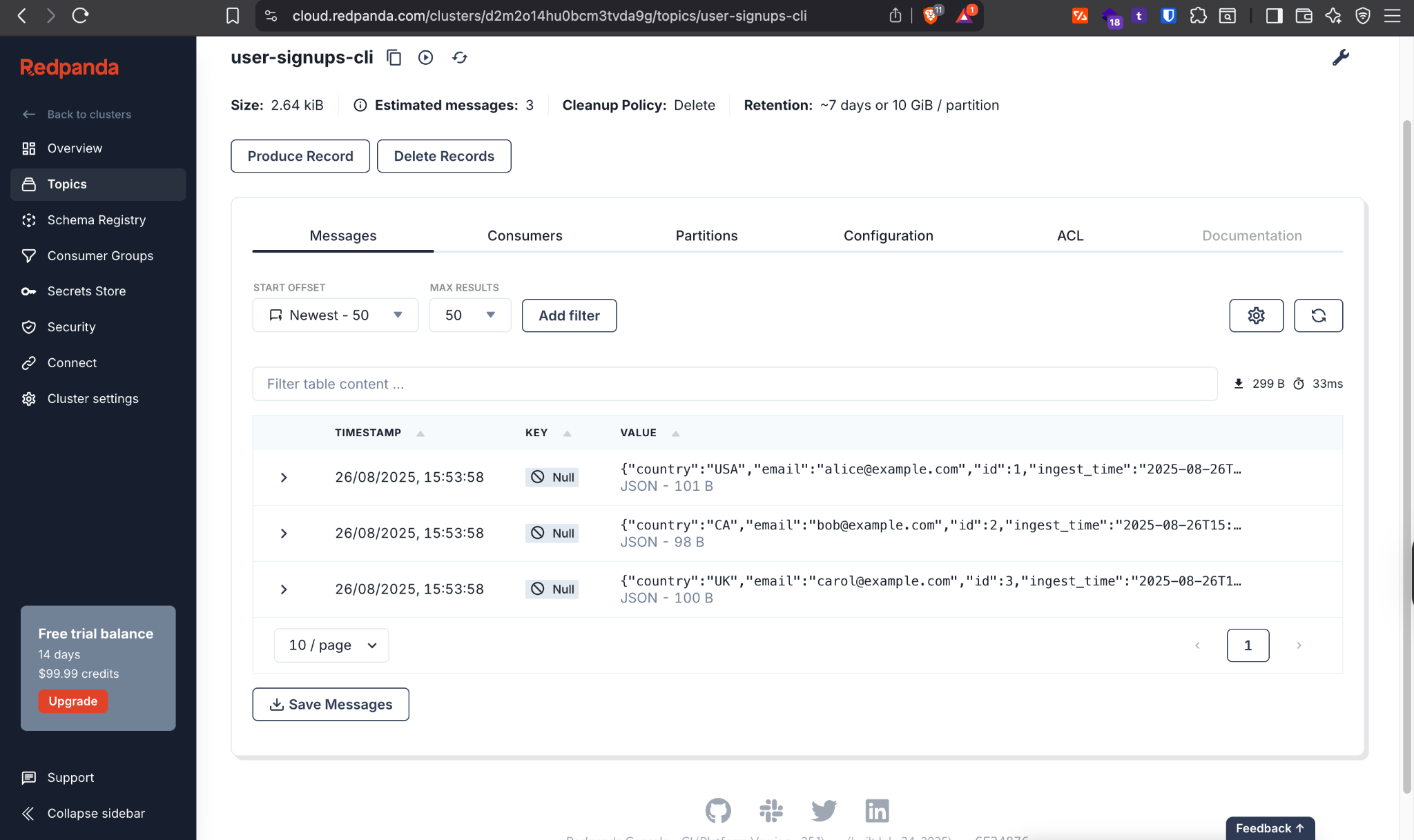
## **Step 6: Verify the Ingestion**

1. **Verify with rpk:** Use rpk to confirm the data has landed in your cloud topic.

|  |
| --- |
| rpk topic consume user-signups-cli -n 3 --profile rpk-cloud |

You should see the three user records, now enriched with an ingest\_time field.

1. **Verify in Cloud UI:** Navigate to your topic's **Messages** tab in the Redpanda Cloud UI to see the records.



# **Part 2: Streaming Data in Real-time**

## **Step 7: Create the Streaming Pipeline Configuration**

Create a second configuration file named cloud-to-stdout.yaml. This connector will subscribe to your cloud topic and print any messages it receives.

**cloud-to-stdout.yaml**

|  |
| --- |
| # cloud-to-stdout.yaml input:  kafka:  addresses: [ "${REDPANDA\_BROKERS}" ]  topics: [ "user-signups-cli" ]  consumer\_group: "cli-stdout-printer"  start\_from\_oldest: true  tls: { "enabled": true }  sasl:  mechanism: "SCRAM-SHA-256"  user: "${REDPANDA\_USER}"  password: "${REDPANDA\_PASS}" output:  stdout:  codec: "lines" |

## **Step 8: Run the Streaming Pipeline**

In the same terminal, run the new streaming connector.

|  |
| --- |
| rpk connect run --env-file .env ./cloud-to-stdout.yaml |

**Expected output:** The connector will start, print the three existing messages from the topic, and then wait for new messages.

## 

## **Step 9: Test the Real-time Stream**

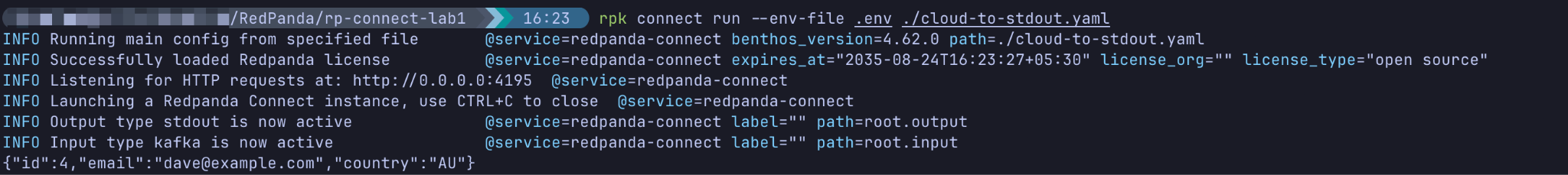
1. **Open a new terminal window** and navigate to the same project directory.
2. Use rpk to produce a new message to your cloud topic.  
   rpk topic produce user-signups-cli --profile rpk-cloud
3. The command will wait for input. Type a new JSON message and press Enter.

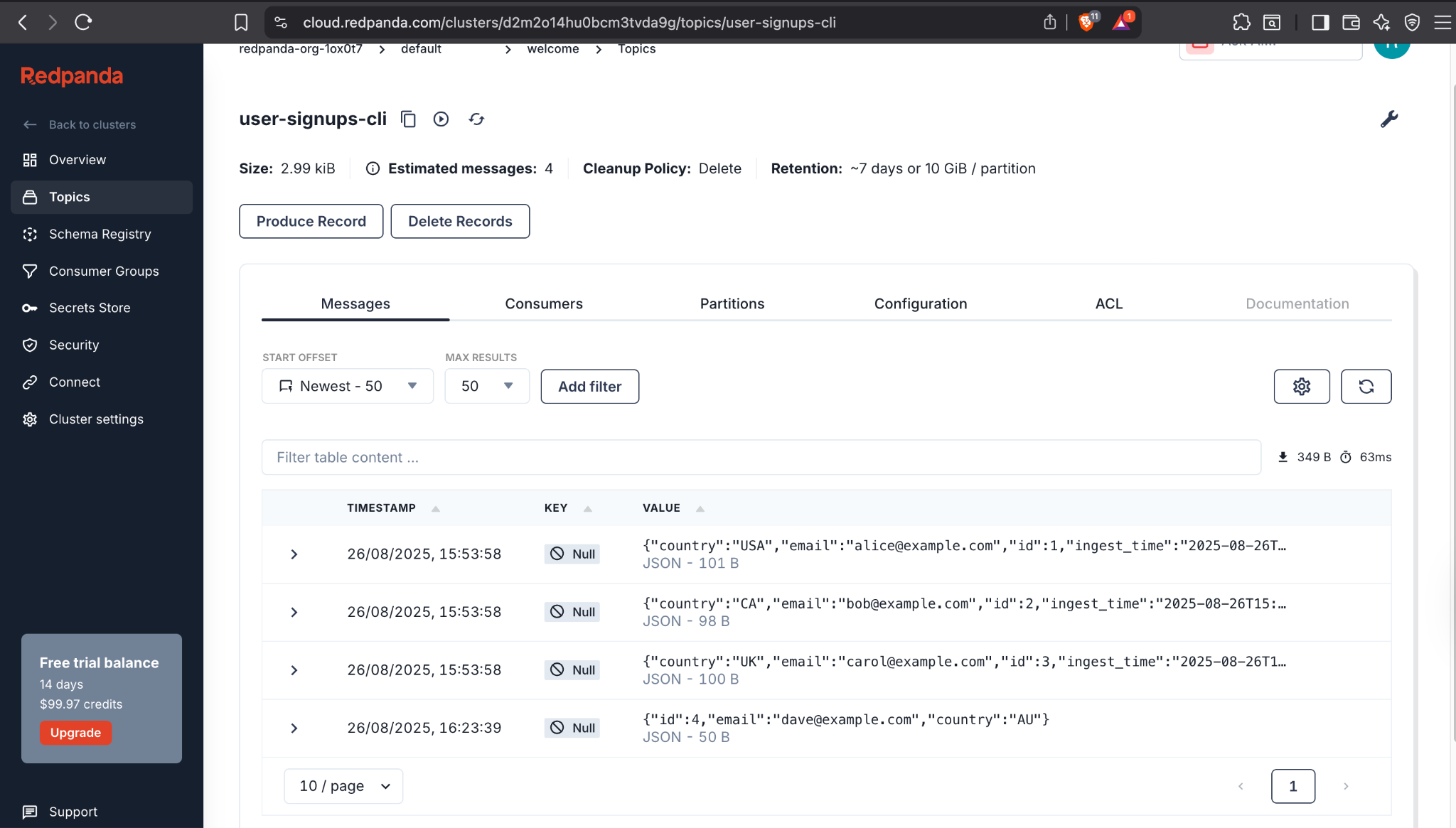
|  |
| --- |
| {"id":4,"email":"dave@example.com","country":"AU"} |

## 

## **Step 10: Verify the Stream**

1. **Verify in Terminal:** Switch back to your first terminal where the streaming connector is running. You should see the new message for "dave" appear almost instantly.



1. **Verify in Cloud UI:** Refresh the **Messages** tab in the Redpanda Cloud UI. You will now see the fourth record for "dave" at the end of the list.  
   **

# Cleanup

1. Stop the streaming connector by pressing Ctrl+C.
2. Delete the topic from Redpanda Cloud.

|  |
| --- |
| rpk topic delete user-signups-cli --profile rpk-cloud |