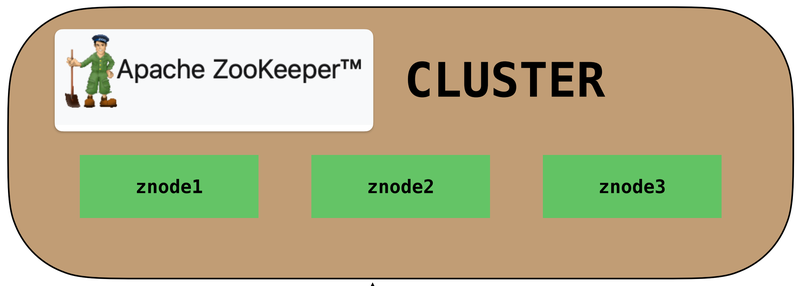
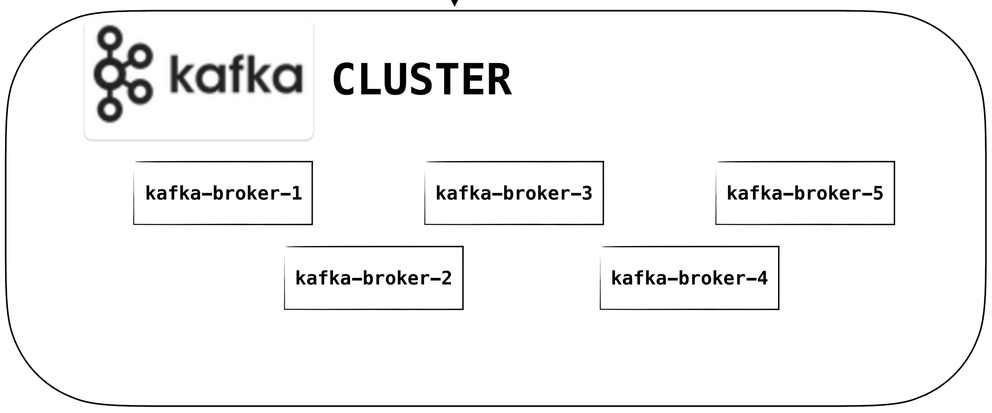


SSL



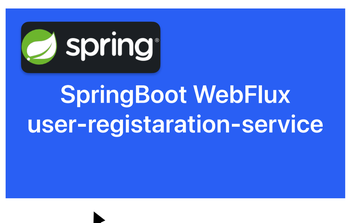
```
{
    "id": 777,
    "customer_id": 1,
    "items": [
        {
            "product_name": "Tea - Earl Grey",
            "product_quantity": 3,
            "product_price": 271.07
        },
        {
            "product_name": "Sausage - Blood Pudding",
            "product_quantity": 3,
            "product_price": 410.74
        }
    ],
    "total": 2045.43,
    "create_date": "1611851587000",
    "shipping_address": "12066 Sheridan Crossing",
    "shipping_city": "Austin",
    "shipping_state": "TX",
    "shipping_zip": "78753"
}
```

SASL_SSL

SASL_SSL

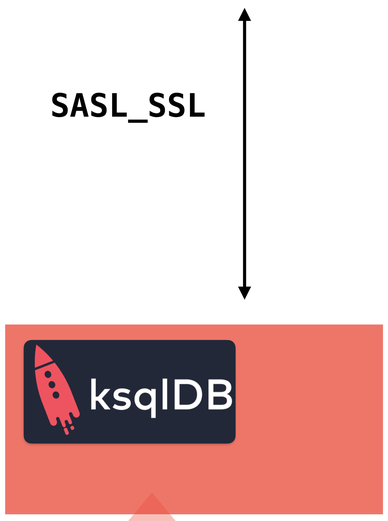
SASL_SSL

SASL_SSL



HTTPS

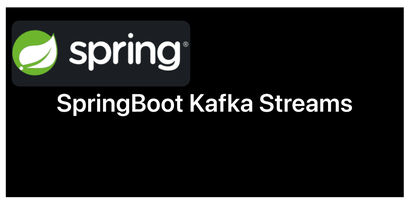
```
{
    "id": 1,
    "first_name": "Nicolas",
    "last_name": "Nunez",
    "email": "nicolas.nunez@toasty.com",
    "dob": "yyyy-mm-dd",
    "country": "Bolivia",
    "city": "La Paz"
}
```

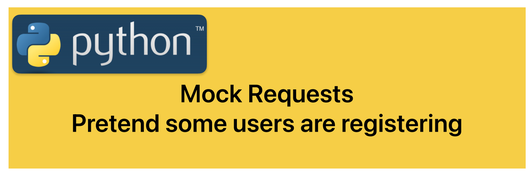



```
KStream<Long, User> userRegistrationKStream = streamsBuilder
        .stream( topic: "user-registration-events");


KTable<String, User> byCountryUserKTable = userRegistrationKStream
        .selectKey((k, v) -> v.getCountry())
        .toTable(Materialized.with(stringSerde, userSerde));

KTable<String, Long> countByCountryKTable = byCountryUserKTable
        .toStream() KStream<String, User>
        .groupByKey() KGroupedStream<String, User>
        .count(Named.as( name: "registrations-by-country"),
                Materialized.with(stringSerde, longSerde));
```






```
CREATE STREAM user_registration_events_stream
(id BIGINT,
first_name VARCHAR,
last_name VARCHAR,
email VARCHAR,
dob DATE,
country VARCHAR,
city VARCHAR)
WITH(kafka_topic='user-registration-events', value_format='json', partitions=5);
```

```
CREATE STREAM order_events_stream
(id BIGINT,
customer_id BIGINT,
items ARRAY<STRUCT< product_name VARCHAR, product_quantity INT, product_price DOUBLE>>,
total DECIMAL(10,2),
currency VARCHAR,
create_date BIGINT,
shipping_address VARCHAR,
shipping_city VARCHAR,
shipping_state VARCHAR,
shipping_zip VARCHAR)
WITH(kafka_topic='order-events', value_format='json', partitions=5);
```

```
CREATE STREAM products_stream AS
SELECT
explode(items)->product_name AS name,
explode(items)->product_quantity AS quantity,
explode(items)->product_price AS price
FROM order_events_stream
EMIT CHANGES;
```

```
CREATE TABLE product_demand AS
SELECT name, COUNT(name) AS CNT
FROM products_stream
GROUP BY name
EMIT CHANGES;
```