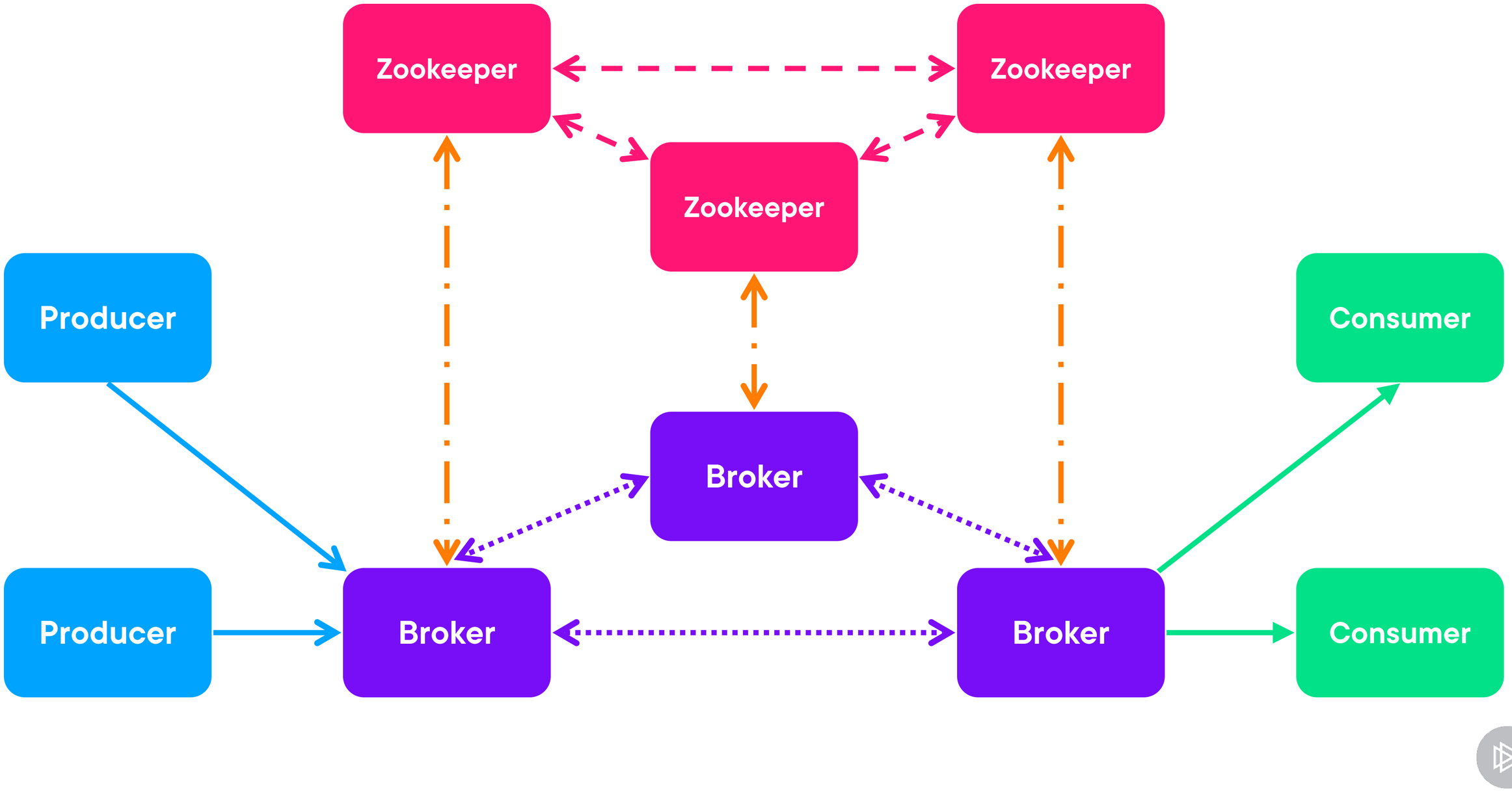
Kafka Administration

Axel Sirota

AI and Cloud Consultant

@AxelSirota


Zookeeper
Zookeeper
Zookeeper
Producer
Producer
Broker
Broker
Broker
Consumer
Consumer

How do I know you are my server?

How do I know this information is untouched?

Certificate Authority (CA)

How do I know you are my server?

Client

Server

How do I know this information is untouched?

TrustStore

Certificate Authority (CA)

KeyStore

How do I know you are my server?

Client

Server

How do I know this information is untouched?

	KeyStore	TrustStore
The CA		
Zookeeper		
Brokers		
Producers		
Consumers		

Creating the KeyStores and TrustStores

Encrypting Zookeeper

Can Be Done in Two Ways

Online Fashion

Offline Fashion

Configuration we need to set in a Zookeeper

```
sslQuorum=true

portUnification=false

serverCnxnFactory=org.apache.zookeeper.server.NettySe
rverCnxnFactory

secureClientPort=2281

ssl.hostnameVerification=false

ssl.quorum.hostnameVerification=false
```

Configuration we need to set in a Zookeeper

```
sslQuorum=true

portUnification=false

serverCnxnFactory=org.apache.zookeeper.server.NettySe
rverCnxnFactory

secureClientPort=2281

ssl.hostnameVerification=false

ssl.quorum.hostnameVerification=false
```

Configuration we need to set in a Zookeeper

```
sslQuorum=true

portUnification=false

serverCnxnFactory=org.apache.zookeeper.server.NettySe
rverCnxnFactory

secureClientPort=2281

ssl.hostnameVerification=false

ssl.quorum.hostnameVerification=false
```

Configuration we need to set in a Zookeeper

```
sslQuorum=true

portUnification=false

serverCnxnFactory=org.apache.zookeeper.server.NettyServerCnxnFactory

secureClientPort=2281

ssl.hostnameVerification=false

ssl.quorum.hostnameVerification=false
```

Configuration we need to set in a Zookeeper

```
sslQuorum=true

portUnification=false

serverCnxnFactory=org.apache.zookeeper.server.NettySe
rverCnxnFactory

secureClientPort=2281

ssl.hostnameVerification=false

ssl.quorum.hostnameVerification=false
```

Configuration we need to set in a Zookeeper

```
sslQuorum=true

portUnification=false

serverCnxnFactory=org.apache.zookeeper.server.NettySe
rverCnxnFactory

secureClientPort=2281

ssl.hostnameVerification=false

ssl.quorum.hostnameVerification=false
```

Finally, we set that we just add the KeyStores and TrustStores as volumes and then in the same configuration

```
ssl.quorum.keyStore.location=/security/
zookeeper-1.keystore.jks

ssl.quorum.keyStore.password=password

ssl.quorum.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.quorum.trustStore.password=password

ssl.keyStore.location=/security/zookeeper-1.keystore.jks

ssl.keyStore.password=password

ssl.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.trustStore.password=password
```

Finally, we set that we just add the KeyStores and TrustStores as volumes and then in the same configuration

```
ssl.quorum.keyStore.location=/security/
zookeeper-1.keystore.jks

ssl.quorum.keyStore.password=password

ssl.quorum.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.quorum.trustStore.password=password

ssl.keyStore.location=/security/zookeeper-1.keystore.jks

ssl.keyStore.password=password

ssl.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.trustStore.password=password
```

Finally, we set that we just add the KeyStores and TrustStores as volumes and then in the same configuration

```
ssl.quorum.keyStore.location=/security/
zookeeper-1.keystore.jks

ssl.quorum.keyStore.password=password

ssl.quorum.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.quorum.trustStore.password=password

ssl.keyStore.location=/security/zookeeper-1.keystore.jks

ssl.keyStore.password=password

ssl.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.trustStore.password=password
```

Finally, we set that we just add the KeyStores and TrustStores as volumes and then in the same configuration

```
ssl.quorum.keyStore.location=/security/
zookeeper-1.keystore.jks

ssl.quorum.keyStore.password=password

ssl.quorum.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.quorum.trustStore.password=password

ssl.keyStore.location=/security/zookeeper-1.keystore.jks

ssl.keyStore.password=password

ssl.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.trustStore.password=password
```

Finally, we set that we just add the KeyStores and TrustStores as volumes and then in the same configuration

```
ssl.quorum.keyStore.location=/security/
zookeeper-1.keystore.jks

ssl.quorum.keyStore.password=password

ssl.quorum.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.quorum.trustStore.password=password

ssl.keyStore.location=/security/zookeeper-1.keystore.jks

ssl.keyStore.password=password

ssl.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.trustStore.password=password
```

Finally, we set that we just add the KeyStores and TrustStores as volumes and then in the same configuration

```
ssl.quorum.keyStore.location=/security/
zookeeper-1.keystore.jks

ssl.quorum.keyStore.password=password

ssl.quorum.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.quorum.trustStore.password=password

ssl.keyStore.location=/security/zookeeper-1.keystore.jks

ssl.keyStore.password=password

ssl.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.trustStore.password=password
```

Finally, we set that we just add the KeyStores and TrustStores as volumes and then in the same configuration

```
ssl.quorum.keyStore.location=/security/
zookeeper-1.keystore.jks

ssl.quorum.keyStore.password=password

ssl.quorum.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.quorum.trustStore.password=password

ssl.keyStore.location=/security/zookeeper-1.keystore.jks

ssl.keyStore.password=password

ssl.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.trustStore.password=password
```

Finally, we set that we just add the KeyStores and TrustStores as volumes and then in the same configuration

```
ssl.quorum.keyStore.location=/security/
zookeeper-1.keystore.jks

ssl.quorum.keyStore.password=password

ssl.quorum.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.quorum.trustStore.password=password

ssl.keyStore.location=/security/zookeeper-1.keystore.jks

ssl.keyStore.password=password

ssl.trustStore.location=/security/
zookeeper-1.truststore.jks

ssl.trustStore.password=password
```

We add the TrustStores and KeyStores as volumes and set the environment

```
KAFKA_ZOOKEEPER_CONNECT:
zookeeper-1:2281,zookeeper-2:2281,zookeeper-3:2281

KAFKA_ZOOKEEPER_SSL_CLIENT_ENABLE: "true“

KAFKA_ZOOKEEPER_CLIENT_CNXN_SOCKET:org.apache.zookeeper.ClientCnxnSocketNetty

KAFKA_ZOOKEEPER_SSL_KEYSTORE_LOCATION: /kafka/security/broker-1.keystore.jks

KAFKA_ZOOKEEPER_SSL_KEYSTORE_PASSWORD: password

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_LOCATION: /kafka/security/broker-1.truststore.jks

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_PASSWORD: password
```

We add the TrustStores and KeyStores as volumes and set the environment

```
KAFKA_ZOOKEEPER_CONNECT:
zookeeper-1:2281,zookeeper-2:2281,zookeeper-3:2281

KAFKA_ZOOKEEPER_SSL_CLIENT_ENABLE: "true“

KAFKA_ZOOKEEPER_CLIENT_CNXN_SOCKET:org.apache.zookeeper.ClientCnxnSocketNetty

KAFKA_ZOOKEEPER_SSL_KEYSTORE_LOCATION: /kafka/security/broker-1.keystore.jks

KAFKA_ZOOKEEPER_SSL_KEYSTORE_PASSWORD: password

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_LOCATION: /kafka/security/broker-1.truststore.jks

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_PASSWORD: password
```

We add the TrustStores and KeyStores as volumes and set the environment

```
KAFKA_ZOOKEEPER_CONNECT:
zookeeper-1:2281,zookeeper-2:2281,zookeeper-3:2281

KAFKA_ZOOKEEPER_SSL_CLIENT_ENABLE: "true“

KAFKA_ZOOKEEPER_CLIENT_CNXN_SOCKET:org.apache.zookeeper.ClientCnxnSocketNetty

KAFKA_ZOOKEEPER_SSL_KEYSTORE_LOCATION: /kafka/
security/broker-1.keystore.jks

KAFKA_ZOOKEEPER_SSL_KEYSTORE_PASSWORD: password

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_LOCATION: /kafka/
security/broker-1.truststore.jks

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_PASSWORD: password
```

We add the TrustStores and KeyStores as volumes and set the environment

```
KAFKA_ZOOKEEPER_CONNECT:
zookeeper-1:2281,zookeeper-2:2281,zookeeper-3:2281

KAFKA_ZOOKEEPER_SSL_CLIENT_ENABLE: "true“

KAFKA_ZOOKEEPER_CLIENT_CNXN_SOCKET:org.apache.zookeeper.ClientCnxnSocketNetty

KAFKA_ZOOKEEPER_SSL_KEYSTORE_LOCATION: /kafka/security/broker-1.keystore.jks

KAFKA_ZOOKEEPER_SSL_KEYSTORE_PASSWORD: password

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_LOCATION: /kafka/security/broker-1.truststore.jks

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_PASSWORD: password
```

We add the TrustStores and KeyStores as volumes and set the environment

```
KAFKA_ZOOKEEPER_CONNECT:
zookeeper-1:2281,zookeeper-2:2281,zookeeper-3:2281

KAFKA_ZOOKEEPER_SSL_CLIENT_ENABLE: "true“

KAFKA_ZOOKEEPER_CLIENT_CNXN_SOCKET:org.apache.zookeeper.ClientCnxnSocketNetty

KAFKA_ZOOKEEPER_SSL_KEYSTORE_LOCATION: /kafka/security/broker-1.keystore.jks

KAFKA_ZOOKEEPER_SSL_KEYSTORE_PASSWORD: password

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_LOCATION: /kafka/security/broker-1.truststore.jks

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_PASSWORD: password
```

We add the TrustStores and KeyStores as volumes and set the environment

```
KAFKA_ZOOKEEPER_CONNECT:
zookeeper-1:2281,zookeeper-2:2281,zookeeper-3:2281

KAFKA_ZOOKEEPER_SSL_CLIENT_ENABLE: "true“

KAFKA_ZOOKEEPER_CLIENT_CNXN_SOCKET:org.apache.zookeeper.ClientCnxnSocketNetty

KAFKA_ZOOKEEPER_SSL_KEYSTORE_LOCATION: /kafka/security/broker-1.keystore.jks

KAFKA_ZOOKEEPER_SSL_KEYSTORE_PASSWORD: password

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_LOCATION: /kafka/security/broker-1.truststore.jks

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_PASSWORD: password
```

We add the TrustStores and KeyStores as volumes and set the environment

```
KAFKA_ZOOKEEPER_CONNECT:
zookeeper-1:2281,zookeeper-2:2281,zookeeper-3:2281

KAFKA_ZOOKEEPER_SSL_CLIENT_ENABLE: "true“

KAFKA_ZOOKEEPER_CLIENT_CNXN_SOCKET:org.apache.zookeeper.ClientCnxnSocketNetty

KAFKA_ZOOKEEPER_SSL_KEYSTORE_LOCATION: /kafka/security/broker-1.keystore.jks

KAFKA_ZOOKEEPER_SSL_KEYSTORE_PASSWORD: password

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_LOCATION: /kafka/security/broker-1.truststore.jks

KAFKA_ZOOKEEPER_SSL_TRUSTSTORE_PASSWORD: password
```

Encrypting Zookeepers

Encrypting Producers and Consumers


HOST
LOCALHOST:9092
PLAINTEXT

There Are Four Supported Protocols

PLAINTEXT

SSL

SASL_PLAINTEXT

SASL_SSL

Declaring a listener is quite simple as well, in the environment we need to set

```
KAFKA_LISTENERS: HOST_PLAINTEXT://broker-1:9091

KAFKA_ADVERTISED_LISTENERS: HOST_PLAINTEXT://
broker-1:9091

KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
HOST_PLAINTEXT:PLAINTEXT
```

Declaring a listener is quite simple as well, in the environment we need to set

```
KAFKA_LISTENERS: HOST_PLAINTEXT://broker-1:9091

KAFKA_ADVERTISED_LISTENERS: HOST_PLAINTEXT://
broker-1:9091

KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
HOST_PLAINTEXT:PLAINTEXT
```

Declaring a listener is quite simple as well, in the environment we need to set

```
KAFKA_LISTENERS: HOST_PLAINTEXT://broker-1:9091

KAFKA_ADVERTISED_LISTENERS: HOST_PLAINTEXT://
broker-1:9091

KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
HOST_PLAINTEXT:PLAINTEXT
```

We need to change those to

```
KAFKA_LISTENERS: SSL://broker-1:9192

KAFKA_ADVERTISED_LISTENERS: SSL://broker-1:9192

KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: SSL:SSL

KAFKA_SECURITY_INTER_BROKER_PROTOCOL: SSL

KAFKA_SSL_CLIENT_AUTH: none
```

We need to change those to

```
KAFKA_LISTENERS: SSL://broker-1:9192

KAFKA_ADVERTISED_LISTENERS: SSL://broker-1:9192

KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: SSL:SSL

KAFKA_SECURITY_INTER_BROKER_PROTOCOL: SSL

KAFKA_SSL_CLIENT_AUTH: none
```

We need to change those to

```
KAFKA_LISTENERS: SSL://broker-1:9192

KAFKA_ADVERTISED_LISTENERS: SSL://broker-1:9192

KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: SSL:SSL

KAFKA_SECURITY_INTER_BROKER_PROTOCOL: SSL

KAFKA_SSL_CLIENT_AUTH: none
```

We need to change those to

```
KAFKA_LISTENERS: SSL://broker-1:9192

KAFKA_ADVERTISED_LISTENERS: SSL://broker-1:9192

KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: SSL:SSL

KAFKA_SECURITY_INTER_BROKER_PROTOCOL: SSL

KAFKA_SSL_CLIENT_AUTH: none
```

We need to change those to

```
KAFKA_LISTENERS: SSL://broker-1:9192

KAFKA_ADVERTISED_LISTENERS: SSL://broker-1:9192

KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: SSL:SSL

KAFKA_SECURITY_INTER_BROKER_PROTOCOL: SSL

KAFKA_SSL_CLIENT_AUTH: none
```

Setting the TrustStores and KeyStores

```
KAFKA_SSL_KEYSTORE_LOCATION: /kafka/security/
broker-1.keystore.jks

KAFKA_SSL_KEYSTORE_PASSWORD: password

KAFKA_SSL_KEY_PASSWORD: password

KAFKA_SSL_TRUSTSTORE_LOCATION: /kafka/security/
broker-1.truststore.jks

KAFKA_SSL_TRUSTSTORE_PASSWORD: password
```

Setting the TrustStores and KeyStores

```
KAFKA_SSL_KEYSTORE_LOCATION: /kafka/security/
broker-1.keystore.jks

KAFKA_SSL_KEYSTORE_PASSWORD: password

KAFKA_SSL_KEY_PASSWORD: password

KAFKA_SSL_TRUSTSTORE_LOCATION: /kafka/security/
broker-1.truststore.jks

KAFKA_SSL_TRUSTSTORE_PASSWORD: password
```

Setting the TrustStores and KeyStores

```
KAFKA_SSL_KEYSTORE_LOCATION: /kafka/security/
broker-1.keystore.jks

KAFKA_SSL_KEYSTORE_PASSWORD: password

KAFKA_SSL_KEY_PASSWORD: password

KAFKA_SSL_TRUSTSTORE_LOCATION: /kafka/security/
broker-1.truststore.jks

KAFKA_SSL_TRUSTSTORE_PASSWORD: password
```

Setting the TrustStores and KeyStores

```
KAFKA_SSL_KEYSTORE_LOCATION: /kafka/security/
broker-1.keystore.jks

KAFKA_SSL_KEYSTORE_PASSWORD: password

KAFKA_SSL_KEY_PASSWORD: password

KAFKA_SSL_TRUSTSTORE_LOCATION: /kafka/security/
broker-1.truststore.jks

KAFKA_SSL_TRUSTSTORE_PASSWORD: password
```

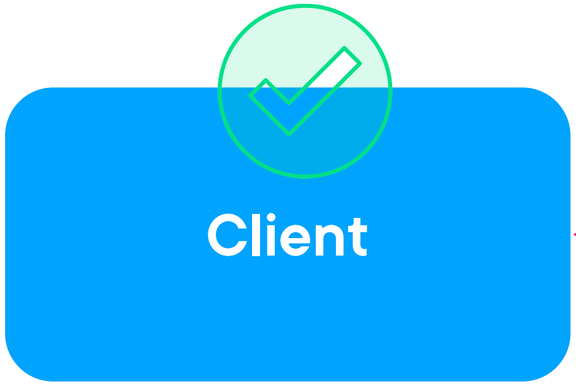
Setting the TrustStores and KeyStores

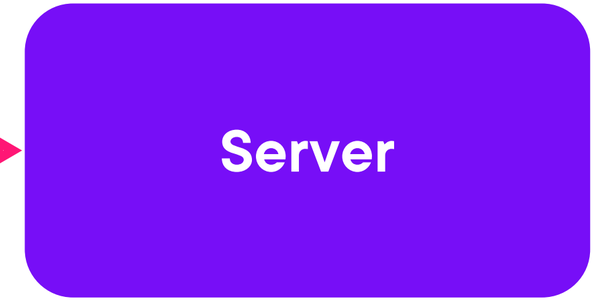
```
KAFKA_SSL_KEYSTORE_LOCATION: /kafka/security/
broker-1.keystore.jks

KAFKA_SSL_KEYSTORE_PASSWORD: password

KAFKA_SSL_KEY_PASSWORD: password

KAFKA_SSL_TRUSTSTORE_LOCATION: /kafka/security/
broker-1.truststore.jks

KAFKA_SSL_TRUSTSTORE_PASSWORD: password
```

Encrypting Producer and Consumer

Authenticating Clients with mTLS

I don’t know how you are





Certificate Authority (CA)

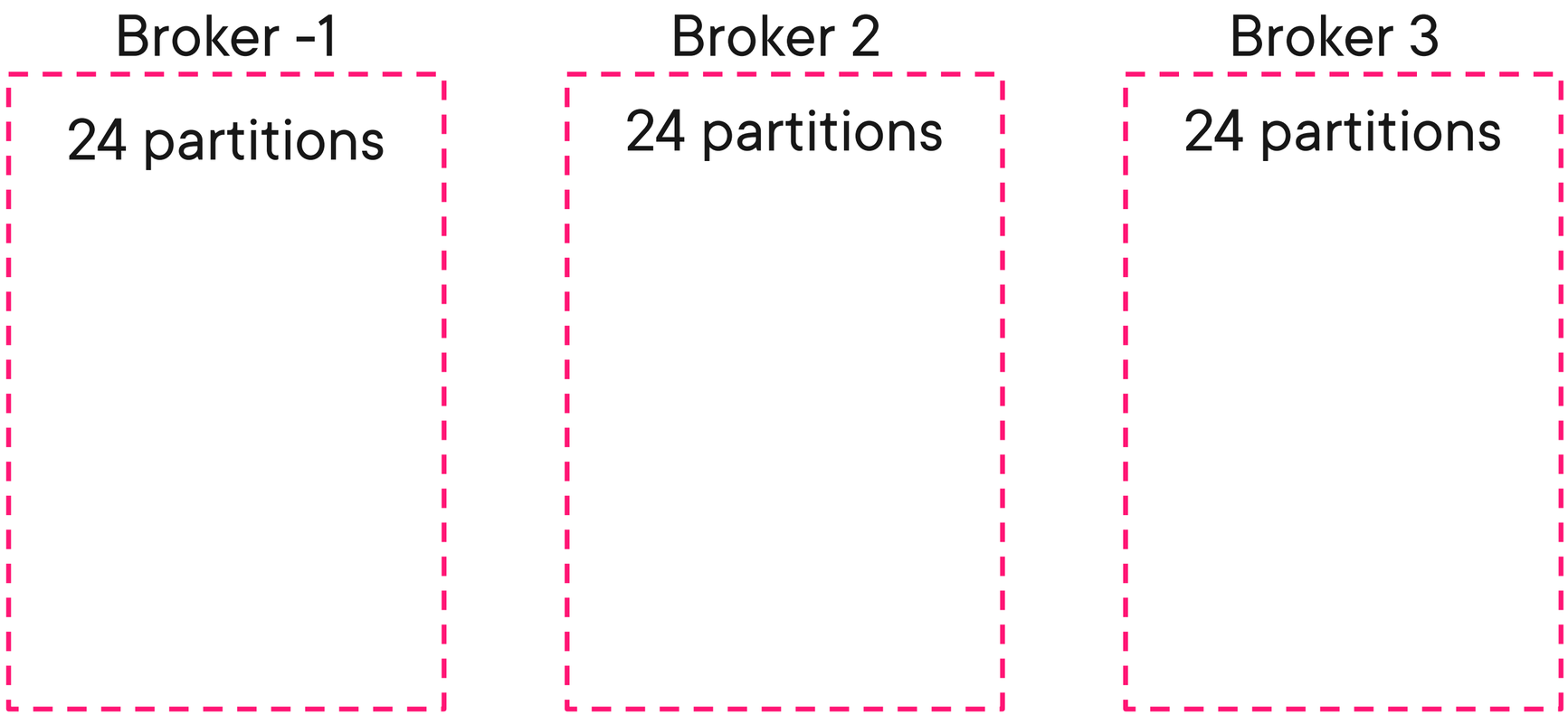
KAFKA_SSL_CLIENT_AUTH: required

don't know WHO you are

Demo

Authenticating Clients

Monitoring Kafka: Scaling


Broker -1
24 partitions
Broker 2
24 partitions
Broker 3
24 partitions

confluent.balancer.enable = true

CBS on

Broker -1

24 partitions

Load

CBS on

Broker 2

24 partitions

Load

CBS on

Broker 3

24 partitions

Load

CBS on

Broker 4

24 partitions

Load

confluent.balancer.enable = true

CBS on

Broker -1

24 partitions

Load

CBS on

Broker 2

24 partitions

Load

CBS on

Broker 3

24 partitions

Load

CBS on

Broker 4

24 partitions

Load

CBS on

Broker -1

24 partitions

Errors

CBS on

Broker 2

24 partitions

Errors

CBS on

Broker 3

24 partitions

Errors

CBS on

Broker 4

24 partitions

Errors

CBS on

Broker -1

24 partitions

CBS on

Broker 2

24 partitions

Errors

CBS on

Broker 3

24 partitions

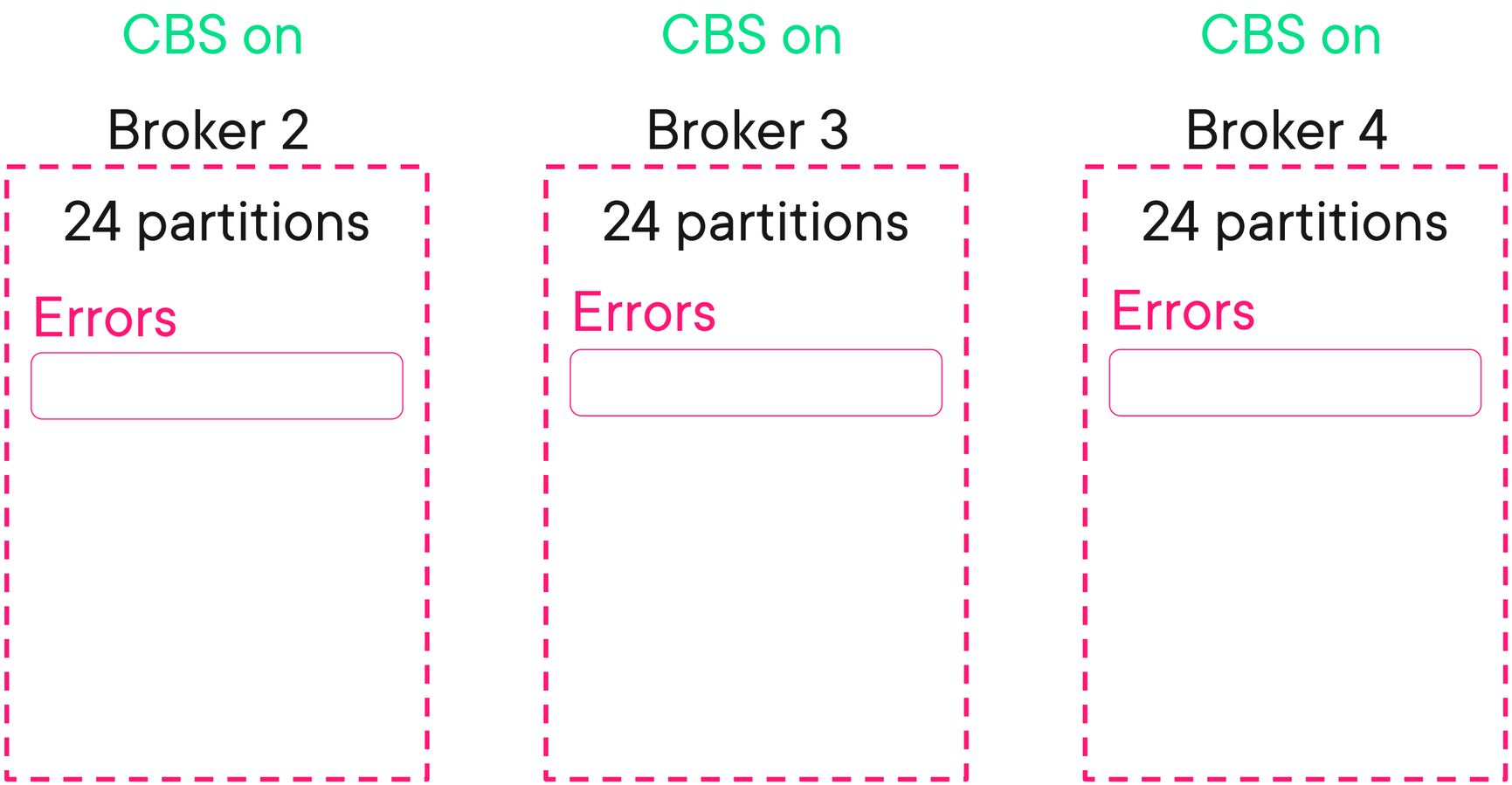
Errors

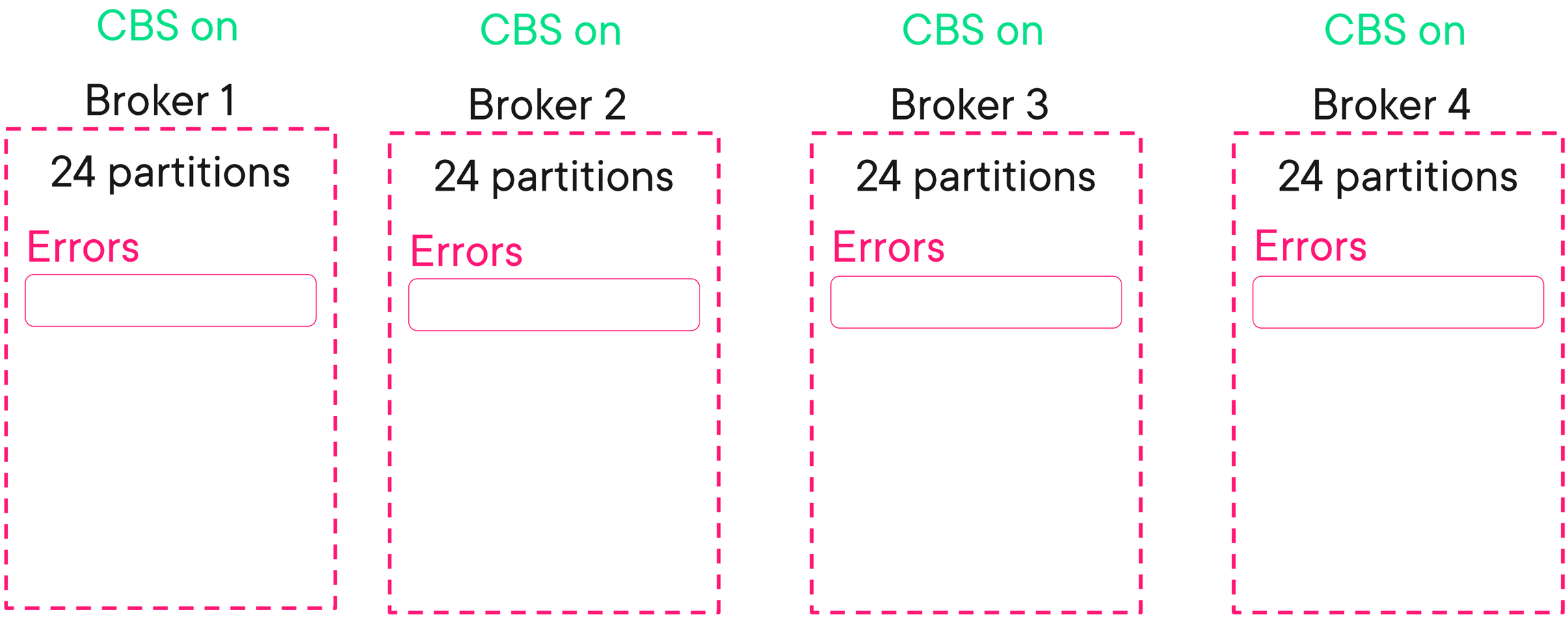
CBS on

Broker 4

24 partitions

Errors


CBS on
Broker 2
24 partitions
Errors
CBS on
Broker 3
24 partitions
Errors
CBS on
Broker 4
24 partitions
Errors

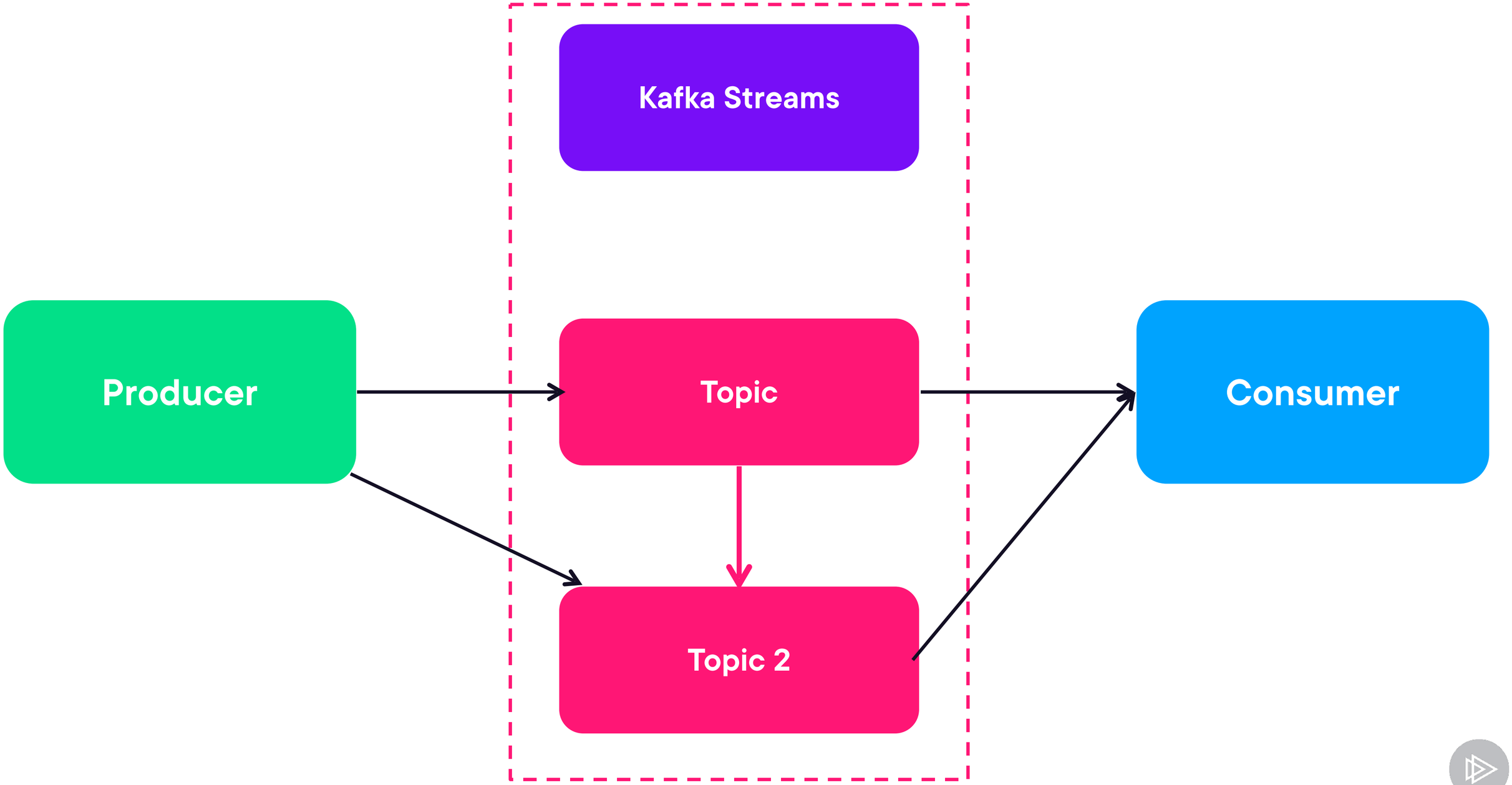

CBS on
Broker 1
24 partitions
Errors
CBS on
Broker 2
24 partitions
Errors
CBS on
Broker 3
24 partitions
Errors
CBS on
Broker 4
24 partitions
Errors

To detect that the command to run is

```
bin/kafka-consumer-groups.sh \
              --bootstrap-server localhost:9092 \
              --describe --group my-group
```

And the output will be like the following

```
TOPIC   PARTITION   CURRENT-OFFSET   LOG-END-OFFSET
LAG   CONSUMER-ID   HOST   CLIENT-ID
my-topic    0               2                4
2     consumer-1-... /127.0.0.1   consumer-1

my-topic    1               2                3
1     consumer-1-... /127.0.0.1   consumer-1

my-topic    2               2                3
1     consumer-2-... /127.0.0.1   consumer-2
```


Kafka Streams
Producer
Topic
Consumer
Topic 2

Key, Takeaways, and Tips

Takeaways

We need to secure all communications channels in Kafka by encrypting with TLS

For that we created TrustStores for all entities and KeyStores for Brokers and Zookeepers

To Authenticate with mTLS we also need a KeyStore for producers and consumers

To scale out Kafka, most of the problems are solved by the Confluent tool, SBC

Keys

Try to do as we did and encrypt and authenticate with mTLS to Kafka

Try to add a new broker and detect the SBC kicking in with the REST Proxy

Try to perform the strategy we spoke about to migrate to a topic with more partitions to scale it out

What Comes Next?

Building ETL Pipelines from Streaming Data with Kafka by Eugene Meidinger

Kafka Connect Fundamentals by Bogdan Sucaciu

Enforcing Data Contracts with Kafka Schema Registry by Bogdan Sucaciu

Securing a Kafka Cluster by Bogdan Sucaciu