Sainsburys PoC Feb25

PoC cluster

- Single node Kraft+Kafka combined mode
  - CFK does not currently provide first class support for KRaft combined mode => use overrides
  - Start from the Kraftcontroller CRD - Kafka CRD requires a 3 node Kraft cluster under CFK
- Self-signed TLS - auto-generated certs
  - SASL_PLAINTEXT inside the pod (for CONTROLLER and REPLICATION listeners) - otherwise need to specify FQDNs for host names (or add localhost to SANs...)
- File-based user creds store - SASL_SSL with Basic creds for external AuthN
- Kafka ACLs
- JMX exporter for Prometheus
- Persistent storage - (but depends on a suitable Storage class)

Add basic authN and TLS

Secret - credential as user store. On pod:

```
cat /mnt/secrets/credential/plain-users.json

# Generate a CA pair for auto generated certs
TUTORIAL_HOME=/Users/nmiddleton/code-local/oauth-cluster/

     openssl genrsa -out $TUTORIAL_HOME/ca-key.pem 2048

     openssl req -new -key $TUTORIAL_HOME/ca-key.pem -x509 \
       -days 1000 \
       -out $TUTORIAL_HOME/ca.pem \
       -subj "/C=US/ST=CA/L=MountainView/O=Confluent/
OU=Operator/CN=TestCA"

     kubectl create secret tls ca-pair-sslcerts \
       --cert=$TUTORIAL_HOME/ca.pem \
       --key=$TUTORIAL_HOME/ca-key.pem
```

Test client connections on kafka pod

Add ACLs with a superuser - kafka

```
cat <<-EOF > /tmp/sslcli.properties

bootstrap.servers=kafka.sainsburys.svc.cluster.local:9092

sasl.jaas.config=org.apache.kafka.common.security.plain.Plai
```

```
nLoginModule required username=kafka password=kafka-secret;
     sasl.mechanism=PLAIN
     security.protocol=SASL_SSL
     ssl.truststore.location=/mnt/sslcerts/truststore.jks
     ssl.truststore.password=mystorepassword
EOF
bootstrap=kafka.sainsburys.svc.cluster.local:9092

kafka-topics --list   --bootstrap-server $bootstrap --
command-config /tmp/sslcli.properties

kafka-acls --bootstrap-server $bootstrap \
--command-config /tmp/sslcli.properties \
--add \
--allow-principal "User:kafka_client" \
--operation All \
--topic '*' \
--group '*'

kafka-acls --bootstrap-server $bootstrap \
 --command-config /tmp/sslcli.properties \
 --list
```

Test non super user client kafka_client

```
cat <<-EOF > /tmp/sslcli.properties

bootstrap.servers=kafka.sainsburys.svc.cluster.local:9092

sasl.jaas.config=org.apache.kafka.common.security.plain.Plai
nLoginModule required username=kafka_client
password=kafka_client-secret;
     sasl.mechanism=PLAIN
     security.protocol=SASL_SSL
     ssl.truststore.location=/mnt/sslcerts/truststore.jks
     ssl.truststore.password=mystorepassword
EOF
bootstrap=kafka.sainsburys.svc.cluster.local:9092

kafka-topics --list   --bootstrap-server $bootstrap --
command-config /tmp/sslcli.properties
```

Monitoring

Prometheus exporter - check metrics are available on the pod:
`curl -s http://kafka-0:7778 | grep -v '# ' | wc -l`
=> default 22k metrics (need to filter)

Prometheus

```
helm upgrade --install demo-test prometheus-community/
prometheus \
 --set alertmanager.persistentVolume.enabled=false \
 --set server.persistentVolume.enabled=false
```

Get the **running** Prometheus server URL by running these commands in the same shell:

```
  export POD_NAME=$(kubectl get pods -l "app.kubernetes.io/
name=prometheus,app.kubernetes.io/component=server" --field-
selector=status.phase=Running -o
jsonpath="{.items[0].metadata.name}")
  kubectl port-forward $POD_NAME 9090
```

http://localhost:9090

Query standard metrics in Prometheus

```
kafka_controller_kafkacontroller_value{name="ActiveControlle
rCount",namespace="sainsburys"}
kafka_server_raft_metrics_current_leader{namespace="sainsbur
ys"}
```

Install Grafana, add dashboards...