Certainly! Below are the detailed commands for securing Kafka topics using basic authentication and authorization with ACLs in a Kubernetes environment with Kraft:

1. \*\*Configure Kafka Server Properties\*\*:

- Modify the Kafka `kafka-config` ConfigMap or directly configure the Kafka pod definition with the following properties:

```yaml

listeners: SASL\_PLAINTEXT://your\_kafka\_server:9092

security.inter.broker.protocol: SASL\_PLAINTEXT

sasl.mechanism.inter.broker.protocol: PLAIN

sasl.enabled.mechanisms: PLAIN

```

2. \*\*Enable Authentication\*\*:

- Set up the necessary JAAS configuration by mounting the JAAS file into the Kafka pod or setting it through environment variables. Here's an example using environment variables:

```yaml

env:

- name: KAFKA\_OPTS

value: "-Djava.security.auth.login.config=/path/to/kafka\_server\_jaas.conf"

```

3. \*\*Create JAAS Configuration File\*\*:

- Create a JAAS configuration file `kafka\_server\_jaas.conf` with the following content:

```

KafkaServer {

org.apache.kafka.common.security.plain.PlainLoginModule required

username="kafkaadmin"

password="adminpassword"

user\_kafka="kafkapassword"

user\_other="otherpassword";

};

```

4. \*\*Configure Producer and Consumer Clients\*\*:

- For each producer and consumer client, configure the JAAS configuration through environment variables or mounted files. Example using environment variables:

```yaml

env:

- name: KAFKA\_OPTS

value: "-Djava.security.auth.login.config=/path/to/kafka\_client\_jaas.conf"

```

5. \*\*Restart Kafka Server and Clients\*\*:

- Apply changes to your Kafka deployment. If using Kubernetes, this typically involves restarting pods or applying changes to ConfigMaps and StatefulSets.

6. \*\*Set Up ACLs for Authorization\*\*:

- Use the Kafka ACL tool or Kafka AdminClient API to set up access control lists for topics. Example commands:

```bash

kafka-acls --authorizer-properties zookeeper.connect=your\_zookeeper\_server:2181 --add --allow-principal User:kafkaadmin --operation Read --topic your\_topic\_name

kafka-acls --authorizer-properties zookeeper.connect=your\_zookeeper\_server:2181 --add --allow-principal User:kafkaadmin --operation Write --topic your\_topic\_name

```

7. \*\*Verify Configuration\*\*:

- Test your configuration by producing and consuming messages from secured topics, ensuring that authentication and authorization are working as expected.

These commands should help you secure your Kafka topics using basic authentication and authorization in a Kubernetes environment with Kraft. Adjust the configurations and paths according to your specific setup and requirements.