[redacted]

# Securing Apache Kafka

# Performance impact with SSL

Tos

- **r3.xlarge**
  - **4 core, 30GB ram, 80GB ssd, moderate network (~90MB/s)**

| | **Throughput(MB/S)** | **CPU on client** | **CPU on broker** |
|---|---|---|---|
| Producer(plaintext) | 83 | 12% | 30% |
| Producer (SSL) | 69 | 28% | 48% |
| Consumer (plaintext) | 83 | 8% | 2% |
| Consumer (SSL) | 69 | 27% | 24% |

- **Most overhead from encryption**

# Configuring SSL

- No client code change; just configuration change.

**Client/Broker**

```
ssl.keystore.location =
/var/private/ssl/kafka.server.keystore.jks
ssl.keystore.password = test1234
ssl.key.password = test1234
ssl.truststore.location =
/var/private/ssl/kafka.server.truststore.jks
ssl.truststore.password = test1234
```

**Broker**

```
listeners = SSL://host.name:port
security.inter.broker.protocol = SSL
ssl.client.auth = required
```

**Client**

```
security.protocol = SSL
```

# Configuring Kerberos

Tos

**No client code change; just configuration change**

### Broker JAAS file

```
KafkaServer {
    com.sun.security.auth.module.
    Krb5LoginModule required
    useKeyTab=true
    storeKey=true
    keyTab="/etc/security/keyt
    abs/kafka_server.keytab"
    principal="kafka/kafka1.ho
    stname.com@EXAMPLE.COM";
};
```

### Client JAAS file

```
KafkaClient {
    com.sun.security.auth.module.
    Krb5LoginModule required
    useKeyTab=true
    storeKey=true
    keyTab="/etc/security/keyt
    abs/kafka_client.keytab"
    principal="kafka-client-
    1@EXAMPLE.COM";
};
```

### Broker JVM

```
Djava.security.auth.lo
gin.config=/etc/kafka/
kafka_server_jaas.conf
```

### ClientJVM

```
-
Djava.security.auth.log
in.config=/etc/kafka/
kafka_client_jaas.conf
```

### Broker config

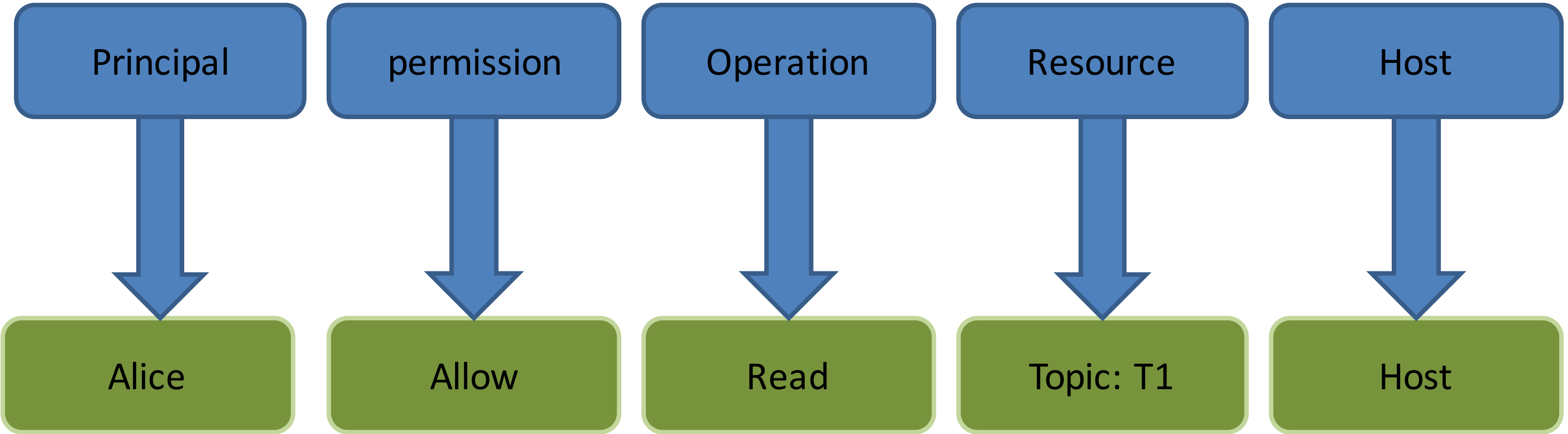
```
security.inter.broker.protocol=
SASL_PLAINTEXT(SASL_SSL)
sasl.kerberos.service.name=kafka
```

### Client config

```
security.protocol=SA
SL_PLAINTEXT(SASL_SSL)
sasl.kerberos.servic
e.name=kafka
```

# ACL

**Alice is Allowed to Read from topic T1 from Host1**



Principal

Alice

permission

Allow

Operation

Read

Resource

Topic: T1

Host

Host

# Operations and Resources

Tos

- Operations
  - Read, Write, Create, Describe, ClusterAction, All
- Resources
  - Topic, Cluster and ConsumerGroup

| **Operations** | **Resources** |
|---|---|
| Read, write, Describe (Read, Write implies Describe) | Topic |
| Read | Consumer Group |
| Create, ClusterAction(communication between controller and brokers) | Cluster |

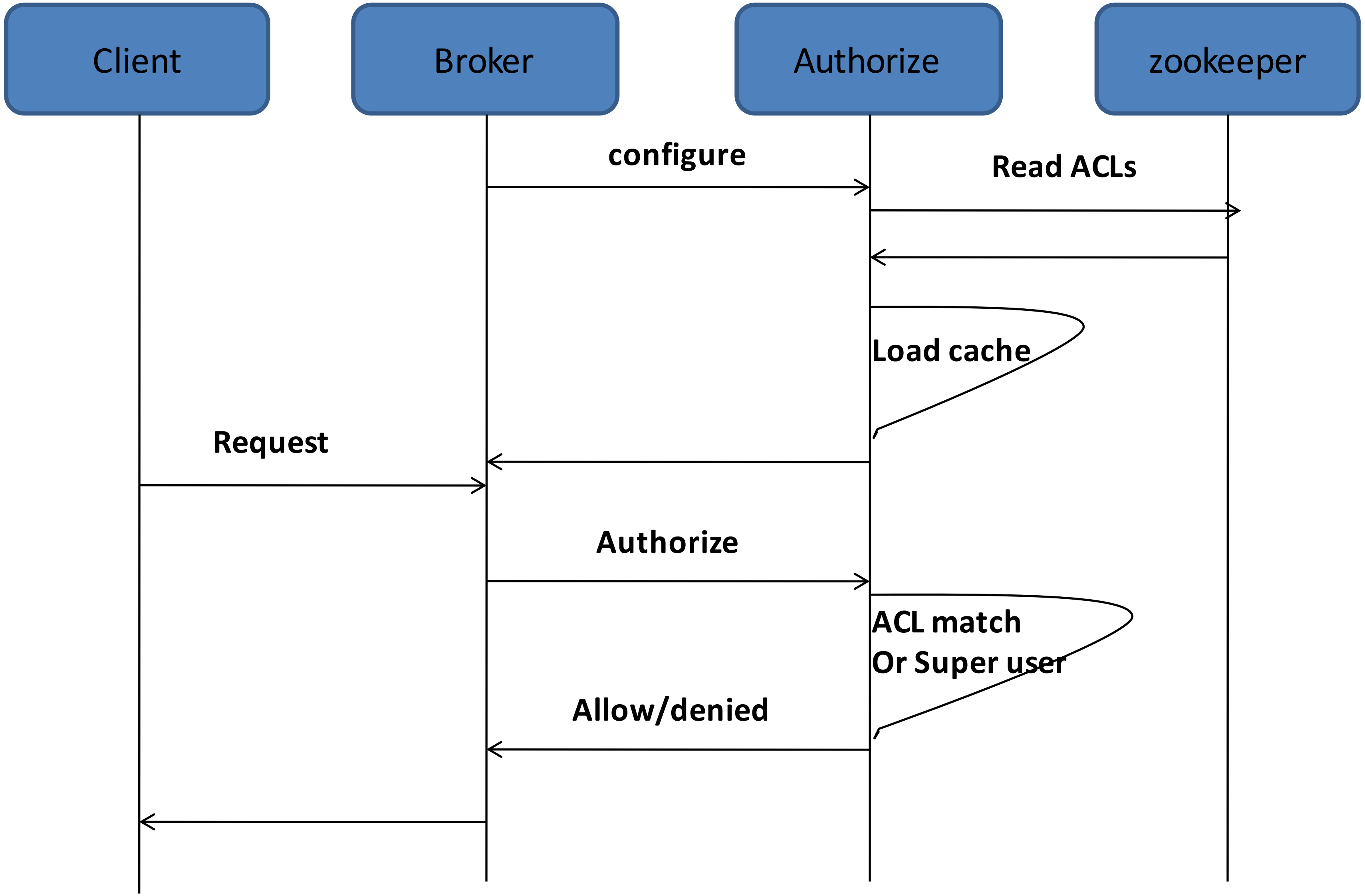
# SimpleAclAuthorizer

Tos

- Out of box authorizer implementation.
- CLI tool for adding/removing acls
- ACLs stored in zookeeper and propagated to brokers asynchronously
- ACL cache in broker for better performance

# Authorizer Flow

Tos



# Configure broker ACL

Tos

- authorizer.class.name=kafka.security.auth.SimpleAclAuthorizer
- Make Kafka principal super users
  - Or grant ClusterAction and Read all topics to Kafka principal

# Configure client ACL

- Producer

  - Grant Write on topic, Create on cluster (auto creation)
  - Or use --producer option in CLI

```
bin/kafka-acls --authorizer-properties zookeeper.connect=localhost:2181 \
     --add --allow-principal User:Bob --producer --topic t1
```

- Consumer

  - Grant Read on topic, Read on consumer group
  - Or use --consumer option in CLI

```
bin/kafka-acls --authorizer-properties zookeeper.connect=localhost:2181 \
    --add --allow-principal User:Bob --consumer --topic t1 --group group1
```

# Lab : - Securing Kafka