package com.twitter.search.ingester.pipeline.wire;

import java.util.concurrent.TimeUnit;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;

import com.google.common.base.Preconditions;

import org.apache.thrift.protocol.TBinaryProtocol;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.twitter.common.quantity.Amount;

import com.twitter.common.quantity.Time;

import com.twitter.common\_internal.manhattan.ManhattanClient;

import com.twitter.common\_internal.manhattan.ManhattanClientImpl;

import com.twitter.finagle.Service;

import com.twitter.finagle.ThriftMux;

import com.twitter.finagle.builder.ClientBuilder;

import com.twitter.finagle.builder.ClientConfig.Yes;

import com.twitter.finagle.mtls.authentication.ServiceIdentifier;

import com.twitter.finagle.mtls.client.MtlsThriftMuxClient;

import com.twitter.finagle.mux.transport.OpportunisticTls;

import com.twitter.finagle.stats.DefaultStatsReceiver;

import com.twitter.finagle.thrift.ClientId;

import com.twitter.finagle.thrift.ThriftClientRequest;

import com.twitter.manhattan.thriftv1.ConsistencyLevel;

import com.twitter.manhattan.thriftv1.ManhattanCoordinator;

import com.twitter.metastore.client\_v2.MetastoreClient;

import com.twitter.metastore.client\_v2.MetastoreClientImpl;

import com.twitter.util.Duration;

public class StratoMetaStoreWireModule {

private WireModule wireModule;

private static final Logger LOG = LoggerFactory.getLogger(StratoMetaStoreWireModule.class);

public StratoMetaStoreWireModule(WireModule wireModule) {

this.wireModule = wireModule;

}

private static final String MANHATTAN\_SD\_ZK\_ROLE =

WireModule.JNDI\_PIPELINE\_ROOT + "manhattanSDZKRole";

private static final String MANHATTAN\_SD\_ZK\_ENV =

WireModule.JNDI\_PIPELINE\_ROOT + "manhattanSDZKEnv";

private static final String MANHATTAN\_SD\_ZK\_NAME =

WireModule.JNDI\_PIPELINE\_ROOT + "manhattanSDZKName";

private static final String MANHATTAN\_APPLICATION\_ID = "ingester\_starbuck";

private static class Options {

// The client id as a string

private final String clientId = "ingester";

// The connection timeout in millis

private final long connectTimeout = 50;

// The request timeout im millis

private final long requestTimeout = 300;

// Total timeout per call (including retries)

private final long totalTimeout = 500;

// The maximum number of retries per call

private final int retries = 2;

}

private final Options options = new Options();

private ClientBuilder<ThriftClientRequest, byte[], ?, Yes, Yes> getClientBuilder(

String name,

ServiceIdentifier serviceIdentifier) {

return getClientBuilder(name, new ClientId(options.clientId), serviceIdentifier);

}

private ClientBuilder<ThriftClientRequest, byte[], ?, Yes, Yes> getClientBuilder(

String name,

ClientId clientId,

ServiceIdentifier serviceIdentifier) {

Preconditions.checkNotNull(serviceIdentifier,

"Can't create Metastore Manhattan client with S2S auth because Service Identifier is null");

LOG.info(String.format("Service identifier for Metastore Manhattan client: %s",

ServiceIdentifier.asString(serviceIdentifier)));

return ClientBuilder.get()

.name(name)

.tcpConnectTimeout(new Duration(TimeUnit.MILLISECONDS.toNanos(options.connectTimeout)))

.requestTimeout(new Duration(TimeUnit.MILLISECONDS.toNanos(options.requestTimeout)))

.timeout(new Duration(TimeUnit.MILLISECONDS.toNanos(options.totalTimeout)))

.retries(options.retries)

.reportTo(DefaultStatsReceiver.get())

.stack(new MtlsThriftMuxClient(ThriftMux.client())

.withMutualTls(serviceIdentifier)

.withClientId(clientId)

.withOpportunisticTls(OpportunisticTls.Required()));

}

/\*\*

\* Returns the Metastore client.

\*/

public MetastoreClient getMetastoreClient(ServiceIdentifier serviceIdentifier)

throws NamingException {

Context jndiContext = new InitialContext();

String destString = String.format("/cluster/local/%s/%s/%s",

jndiContext.lookup(MANHATTAN\_SD\_ZK\_ROLE),

jndiContext.lookup(MANHATTAN\_SD\_ZK\_ENV),

jndiContext.lookup(MANHATTAN\_SD\_ZK\_NAME));

LOG.info("Manhattan serverset Name: {}", destString);

Service<ThriftClientRequest, byte[]> service =

ClientBuilder.safeBuild(getClientBuilder("metastore", serviceIdentifier).dest(destString));

ManhattanClient manhattanClient = new ManhattanClientImpl(

new ManhattanCoordinator.ServiceToClient(service, new TBinaryProtocol.Factory()),

MANHATTAN\_APPLICATION\_ID,

Amount.of((int) options.requestTimeout, Time.MILLISECONDS),

ConsistencyLevel.ONE);

return new MetastoreClientImpl(manhattanClient);

}

}