KIP-967: Support custom SSL configuration for Kafka Connect RestServer

Status

Current state: *Under Discussion*

Discussion thread: here

Voting thread: here

JIRA: Unable to render Jira issues macro, execution error.

Motivation

Add an ability to use custom SSL factory to configure Kafka Connect RestServer.
Currently Kafka Connect provides only one basic mechanism based on file key stores to configure SSL for REST server.
Kafka Connect is used by all sizes of organizations serving varied technical and business domains. SSL/TLS communication is a very critical part of organizations' standards. SSL config customization is the most part of functionality for any applications.

Background

Common approach to configure extensible SSL for Kafka broker & clients was introduced at the KIP-519. The property `ssl.engine.factory.class w` as added in this patch to specify custom creation of the SSL Engine.

Public Interfaces

Use `SSLConfig.SSL_ENGINE_FACTORY_CLASS_CONFIG` (`"ssl.engine.factory.class"`) property with prefixes:

- `"listeners.https."`
- `"admin.listeners.https."`

New config properties are used to define SSL engine factory for RestServer listeners. By default `DefaultSslEngineFactory` is used. Full names of new properties:

- `listeners.https.ssl.engine.factory.class`
- `admin.listeners.https.ssl.engine.factory.class`

All properties prefixed with `"listeners.https."` or `"admin.listeners.https."` passed to the `configure` method of the `SslEngineFactory` instance.

Proposed Changes

There is a public Kafka interface to define custom SSL engine since 2.6.0 version (`SslEngineFactory`). This interface can be used to configure SSL for Kafka connect RestServer.

Add private adapter class to use `SslEngineFactory` for Jetty. Implementation of the main functionality:

SslContextFactoryImpl

```
 class SslContextFactoryServerImpl extends SslContextFactory.Server {
    private final SslEngineFactory sslEngineFactory;

    SslContextFactoryServerImpl(SslEngineFactory sslEngineFactory) {
        this.sslEngineFactory = sslEngineFactory;
    }

    @Override public SSLEngine newSSLEngine() {
        return sslEngineFactory.createServerSslEngine(null, -1);
    }

    @Override public SSLEngine newSSLEngine(String host, int port) {
         return sslEngineFactory.createServerSslEngine(host, port);
    }
}
```

A similar adapter can be used for RestClient (extends `SslContextFactory.Client`).

Compatibility, Deprecation, and Migration Plan

Utilities are affected by the change:

- ConnectStandalone;
- ConnectDistributed;
- MirrorMaker.

There is no impact on existing behavior, and the existing behavior is not deprecated. All existing SSL properties are supported.

Pay an attention that current implementation uses SSL configuration from Kafka client (without any prefixes) for REST servers/client in case there is not any properties with "`listeners.https.`" or "`admin.listeners.https.`" prefixes. An implementation must be backward-compatible with this behavior.

Test Plan

Add integration tests to check:

- RestClient creation (modify `RestForwardingIntegrationTest`);
- Custom SSL engine factory to configure RestServer listeners.

Rejected Alternatives

Use native type for Jetty server in public API

The Jetty server uses extenstions of the class `org.eclipse.jetty.util.ssl.SslContextFactory` to configure SSL for connector.

Disadvantages:

- external dependency in public API (e.g. server implementation may be changed for Kafka Connect);
- new type in public interface;
- new default implementation (because new type must implement `Configurable`) which is minimally different from DefaultSslEngineFactory.